CPU MSX

A REVISTA QUE FAZ O MSX ACONTECER

ANO 3 - N°36 - CR$ 298,00

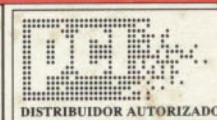
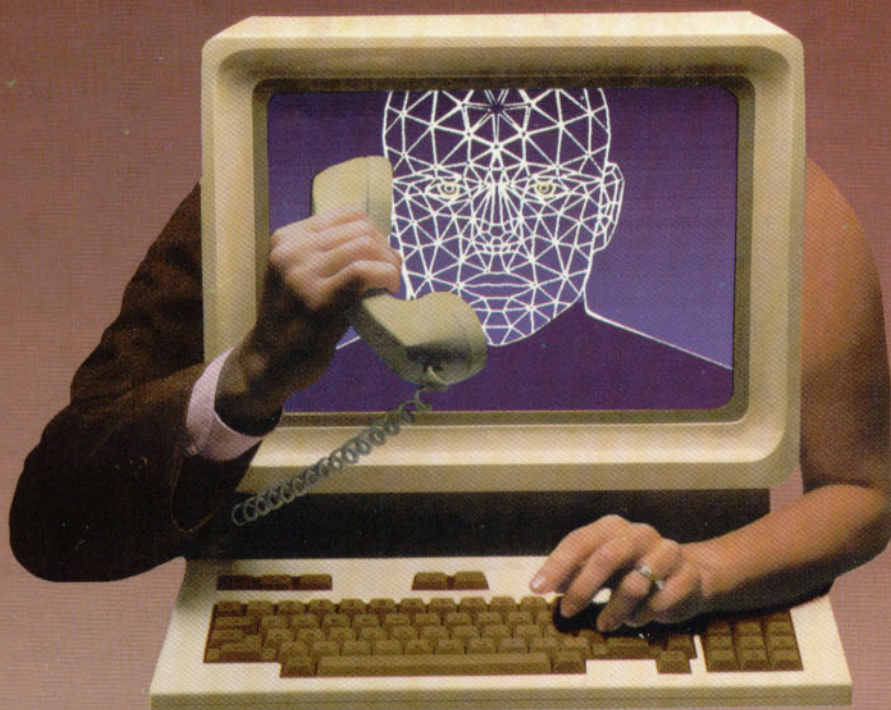
VIA AÉREA: MANAUS, BOA VISTA, SANTARÉM, RIO BRANCO, JIPARANÁ E AMAPÁ

VIDEOPRODUÇÃO NO MSX

CONSTRUA SUA MEGARAM - PARTE 1

DEBUGANDO AS ROTINAS DA ROM - PARTE 2

EDITOR DE RELÓGIOS

JOGO: OPERATION WOLF

MINAS SOFTWARE

ALTA TECNOLOGIA, BONS NEGÓCIOS.

Se a sua empresa é especializada em software, vende equipamentos, presta serviços diversos de informática ou trabalha com o desenvolvimento de cursos e palestras, então não dá para ela ficar de fora deste importante acontecimento, que em outubro vai transferir para Belo Horizonte - Minas - os grandes negócios nacionais envolvendo a área.

Um evento imprescindível para empresários que buscam novos clientes e para clientes em constante procura por soluções de vanguarda.

MINAS SOFTWARE - FÓRUM BRASILEIRO DE SOFTWARE E SERVIÇOS DE INFORMÁTICA

DE 19 A 23 DE OUTUBRO DE 1993 - ÁREA DE EXPOSIÇÃO DO SHOPPING DEL REY.

O seu produto vai estar frente a frente com o público certo.

REALIZAÇÃO

NILSO FARIAS COM. EMP. MG LTDA.

Rua Pio XI, 313 - Ipiranga
31160-140 - Belo Horizonte - MG Brasil
Tel.: 426-2988 - Fax: (031) 426-3086

PATROCÍNIO OFICIAL

APOIO

SHOPPING DEL REY

BÔNUS RIO EDITORA LTDA.
CAIXA POSTAL 11750
CEP 22022-970
RIO DE JANEIRO - RJ
TEL : (021) 255-3667
FAX : (021) 237-6796

DIRETOR EXECUTIVO
JOSE IDEMAR A. NASCIMENTO

JORNALISTA RESPONSÁVEL
DOLAR TANUS
REGISTRO 430-RS

EDITOR TÉCNICO
CARLOS ALBERTO HERSZTERG

CONSULTORES TÉCNICOS
JULIO CESAR SILVA MARCHI
MIGUEL DE ANDRADE FREITAS
ROGÉRIO BELARMINO DA SILVA

ADMINISTRAÇÃO
LUZIMAR GOMES DA SILVA

EDITORAÇÃO ELETRÔNICA E ARTE-FINALIZAÇÃO
JULIO CESAR SILVA MARCHI

REVISÃO
MÁRCIA CHERMAN

PUBLICIDADE
ALEXANDRE MARQUES

ASSINATURAS
LÚCIA HELENA MARCELINO

CAPA
FOCUS INFORMÁTICA

FOTOLITOS
MIL CORES

IMPRESSÃO
GRÁFICA LORD

DISTRIBUIÇÃO
FERNANDO CHINAGLIA DISTRIBUIDORA
R. TEODORO DA SILVA, 907
TEL.: (021) 577-6655

EDITORIAL

Oi pessoal,

Cada vez mais vemos os horizontes do MSX se expandirem em aplicações que escapam do lugar-comum. Definitivamente, já foi-se o tempo em que as máquinas MSX eram basicamente orientadas ao lazer. Hoje podemos ver nossos micros em pequenos estúdios de vídeo executando trabalhos de seqüenciação, titulação, aberturas de vídeo e animação, prestando-se como ferramenta para criação visual, seja ela orientada ao vídeo ou à arte-finalização impressa.

Nesta edição, mostramos um pouco do que é o mundo da videoprodução, suas técnicas e segredos. Para isso, tomamos como configuração mínima um MSX 2. Não podemos escapar dessa realidade – as exigências do mercado ditam o padrão de qualidade.

Além disso, publicamos (finalmente!) a primeira parte do projeto da Megaram, o que irá agradar também aos usuários do MSX 1.

A pedido dos leitores, iniciamos um "revival" dos melhores jogos para MSX. Isto vem uma vez mais confirmar que se por um lado temos leitores que se desenvolveram na informática em um MSX e hoje desejam artigos cada vez mais técnicos ou as últimas novidades em jogos, por outro lado (mas não na contramão) existem aqueles com muito menos tempo de estrada. E continuaremos a publicar artigos, matérias e listagens para esse público mais novo, sem perder a atualidade que a informática exige.

Por fim, gostaria de agradecer (de novo!) a todos que têm colaborado conosco. Graças a esse apoio, podemos pautar as edições seguintes com vários artigos de qualidade – o que nos causa um "problema" enorme para definir qual será a chamada de capa. Contamos com vocês para continuar com esse "problema"!

Carlos Alberto Herszterg

INDICE

Fire Hawk BBS

Já está no ar a mais nova BBS para MSX. A Fire Hawk é a única BBS do país montada sobre uma plataforma MSX 2 com 512 Kb. O software gerenciador, totalmente desenvolvido em Pascal por Fernando Carneiro, SysOp da Fire Hawk, explora com eficiência a velocidade de 1200 bps, disponível na maioria dos modems para MSX, tornado o acesso quatro vezes mais rápido. Para os usuários de PC ou Amiga, está disponível a velocidade de 300 bps.

FIRE HAWK BBS
Tel.: (021) 393-4490
Diariamente das 21h às 6h
SysOp: Fernando Carneiro

Cobra Software e MSX Força — Novos games

Provando que o MSX não morreu, a Cobra Software e a MSX Força se uniram para trazer da Europa as últimas novidades em games para MSX 1 e 2. Com essa iniciativa, as duas empresas já têm em catálogo novos jogos como Magic Johnson (basquete), Power Drift (corrida de carros), The Games Winter Edition (jogos de inverno), Killer Station (espacial), Quasar (espacial), Steve Davis Snooker (sinuca), King Solomons Mines (adventure) e muitos outros. Estes jogos estão sendo comercializados com exclusividade pelas duas empresas e os interessados podem escrever diretamente para:

MSX FORÇA
Rua Pedro Américo, 378/07
Catete – Rio de Janeiro – RJ
Cep 22211-200
Tel.: (021) 265-9265

COBRA SOFTWARE
Rua Chady Muradi, 81
Jaguaré – São Paulo – SP
Cep 05351-050
Tel.: (011) 819-2706

MSX Graphics: Aplicações gráficas no MSX

MSX Graphics é primeira software house voltada para as aplicações gráficas no MSX 2/2+ e Turbo R e está lançando os manuais de operação dos principais programas do mercado como:

- Dynamic Publisher – editoração eletrônica
- Graph Saurus – desenho
- The Final Graph (MSX Paint 4) – desenho e videoprodução
- Video Graphics Philips – videoprodução

Além dos programas e seus respectivos manuais, a empresa possui um completo acervo de fontes de letras, shapes e telas digitalizadas. Os usuários do FM-PAC também encontram jogos e músicas em FM.

MSX GRAPHICS
Caixa Postal 11504
Cep 22022-970
Rio de Janeiro – RJ

MSX 2 e MSX 2+

Debugando a ROM – parte 2

Julio Cesar Silva Marchi
André Luiz Rocha Tupinambá

Como prometido na edição anterior, aqui está o restante das rotinas do BIOS do MSX 2. Para alguns pode até parecer que estamos tentando "encher lingüiça" com estas listagens e depurações, mas gostaríamos de relembrar a importância de informações deste tipo para programadores sérios, que realmente desejam desenvolver softwares de alto nível, e que não conseguem fazê-lo por não possuírem informações técnicas sobre sua máquina, já que isso é coisa que não existe por aqui.

Não há muito mais o que falar; gostaríamos apenas de ressaltar que na próxima edição listaremos as rotinas da SUB-ROM, e, se for possível (por questões de espaço físico), publicaremos os HOOKs que foram incluídos e/ou alterados para a nova versão deste fascinante micro que é o MSX. E não pensem que pára por aí, ainda existe muito mais para ser explorado no interior destas máquinas. Pretendemos fazê-lo sem demagogia neste espaço. Temos recebido e descoberto muitas informações novas sobre os novos MSX e na medida do possível iremos repassando-as para vocês. Esperamos estar assim contribuindo para a manutenção do padrão MSX no Brasil, que, ao contrário do que muitos dizem, continua vivo e forte!.

ROTINAS DA ROM DO MSX 2 - Continuação

• Teclado e saídas padrões

CHSNS (009CH)

Função	: Verifica o buffer de teclado por caracter.
Input	: Nada
Output	: Z se o buffer estiver vazio e NZ caso contrário
Modifica	: AF
Padrão	: 1

CHGET (009FH)

Função	: Espere pela digitação de um caracter
Input	: Nada
Output	: A com o caracter lido
Modifica	: AF
Padrão	: 1

CHPUT (00A2H)

Função	: Manda um caracter para a saída padrão
Input	: A=Caracter a ser impresso
Output	: Nada
Modifica	: Nenhum
Padrão	: 1

LPTOUT (00A5H)

Função	: Manda um caracter para a impressora

ROTINAS DA ROM DO MSX 2 - Continuação

Input : A=Caracter a ser impresso
Output : Nada
Modifica : A Flag C será setada se houver erro
Padrão : 1

LPTSTT (00A8H)
Função : Testa o status da impressora
Input : Nome
Output : Se o registrador A=255 e a Flag Z estiver resetada, a impressora estará Ok; Se o registrador A estiver com 0 e a Flag Z estiver setada, a impres sora não estará pronta;
Modifica : AF
Padrão : 1

CNVCHR (00ABH)
Função : Testa e converte, se necessário, caracteres com cabeçalho gráfico.
Input : A=Caracter
Output : Z,NC=cabeçalho; NZ,C=Gráficos; Z,C=Normal
Modifica : AF
Padrão : 1

PINLIN (00AEH)
Função : Pega uma linha lógica do texto do console
Input : Nada
Output : HL=Início do texto, Flag C caso haja uma interrupção com CTRL+ STOP
Modifica : AF,BC,DE,HL,EI
Padrão : 1

INLIN (00B1H)
Função : O mesmo que PINLIN, só que apresenta o caracter "?" antes de gerar o INPUT.
Input : Nada
Output : HL=Início do texto, Flag C caso haja uma interrupção com CTRL+ STOP
Modifica : AF,BC,DE,HL,EI
Padrão : 1

QINLIN (00B4H)
Função : Executa INLIN sem apresentar o caracter "?"
Input : Nada
Output : HL=Início do texto, Flag C caso haja uma interrupção com CTRL+ STOP
Modifica : AF,BC,DE,HL,EI
Padrão : 1

BREAKX (00B7H)
Função : Testa CTRL+STOP
Input : Nada
Output : Falg C setado se CTRL+STOP for detectado
Modifica : AF
Padrão : 1

BEEP (00C0H)
Função : Gera um BEEP
Input : Nada
Output : Nada
Modifica : Todos
Padrão : 3

CLS (00C3H)
Função : Limpa a tela
Input : Flag Z resetada
Output : Nada
Modifica : AF,BC,DE
Padrão : 3

POSIT (00C6H)
Função : Posiciona o cursor
Input : H com a coordenada X e L com a coordenada Y
Output : Nada
Modifica : AF
Padrão : 1

FNKSB (00C9H)
Função : Testa se as teclas de função estão ativas, se estiverem serão mostradas e caso contrário serão apagadas.
Input : FNKFLG (0FBCEH)
Output : Nada
Modifica : Todos
Padrão : 1

ERAFNK (00CCH)
Função : Apaga as teclas de função
Input : Nada
Output : Nada
Modifica : Todos
Padrão : 1

DSPFNK (00CFH)
Função : Mostra as teclas de função
Input : Nada
Output : Nada
Modifica : Todos
Padrão : 1

TOTEXT (00D2H)
Função : Rotina padrão para devolver o VDP para a Screen 0 (Modos TEXT1 ou TEXT-2) após a execução de um modo gráfico.
Input : Nada
Output : Nada
Modifica : Todos
Padrão : 1

• Acesso de I/O para jogos

GTSTCK (00D5H)
Funçāo : Rotina padrão para devolver o estado do joystick ou das setas do teclado
Input : A - número do joystick

ROTINAS DA ROM DO MSX 2 - Continuação

Output : A - direção do joystick
Modifica : Todos
Padrão : 1

GTTRIG (00D8H)
Função : Rotina padrão para devolver o estado do gatilho do joystick ou da barra de espaço
Input : A - número do gatilho
Output : A - estado 0=solto/ 0FFH=pressionado
Modifica : AF
Padrão : 1

GTPAD (00DBH)
Função : Retorna o estatos do paddle
Input : A - número do paddle
Output : A - valor lido
Modifica : Todos
Padrão : 1

GTPDL (00DEH)
Função : Retorna o valor do paddle
Input : A - número do paddle
Output : A - valor lido
Modifica : Todos
Padrão : 2

• Rotinas de entrada e saída do cassete

TAPION (00E1H)
Função : Liga o motor e lê o cabeçalho do arquivo
Input : Nada
Output : Retorna CARRY se houver erro
Modifica : Todos
Padrão : 1

TAPIN (00E4H)
Função : Rotina para ler um byte do cassete
Input : Nada
Output : A - byte lido, CARRY se houver erro
Modifica : Todos
Padrão : 1

TAPIOF (00E7H)
Função : Interrompe a leitura de dados do cassete
Input : Nada
Output : Nada
Modifica : Nada
Padrão : 1

TAPOON (00EAH)
Função : Liga o motor e escreve o cabeçalho do arquivo
Input : A=0 cabeçalho curto/ A<>0 cabeçalho longo
Output : retorna CARRY se houver erro
Modifica : Todos
Padrão : 1

TAPOUT (00EDH)
Função : Rotina para escrever um byte no cassete
Input : A - byte a ser escrito
Output : Retorna CARRY se houver erro
Modifica : Todos
Padrão : 1

TAPOOF (00F0H)
Função : Rotina para interromper a escrita de dados
Input : Nada
Output : Retorna CARRY se houver erro
Modifica : Nada
Padrão : 1

STMOTR (00F3H)
Função : Seta o modo do cassete
Input : A=0 desliga motor
A=1 liga motor
A=0FFH inverte o estado atual
Output : Nada
Modifica : AF
Padrão : 1

• Miscelâneas

CHGCAP (0132H)
Função : Altera o estado da lâmpada do CAPS LOCK
Input : A=0 apaga/ A<>0 acende
Output : Nada
Modifica : AF
Padrão : 1

CHGSND (0135H)
Função : Altera o bit 1 da porta de status do PSG
Input : A=0 desliga/ A<>0 liga
Output : Nada
Modifica : AF
Padrão : 1

RSLREG (0138H)
Função : Lê o conteúdo da porta de controle dos slots primários da PPI
Input : Nada
Output : A - valor lido
Modifica : A
Padrão : 1

WSLREG (013BH)
Função : Escreve na porta de controle dos slots primários da PPI
Input : A - valor a ser escrito
Output : Nada
Modifica : Nada

ROTINAS DA ROM DO MSX 2 - Continuação

Padrão : 1

RDVDP (013EH)

Função : Lê o conteúdo da porta status do VDP (MSX 1)
Input : Nada
Output : A - valor lido
Modifica : A
Padrão : 1

SNSMAT (0141H)

Função : Lê uma linha da matriz do teclado
Input : A - número da linha
Output : A - valor da linha
Modifica : AF, C
Padrão : 1

ISFLIO (014AH)

Função : Rotina que verifica se o interpretador BASIC está direcionado para algum arquivo
Input : Nada
Output : A=0 ativo/ A<> inativo
Modifica : AF
Padrão : 1

OUTDLP (14DH)

Função : Rotina para imprimir textos formatados
Input : A - byte a ser escrito
Output : nada
Modifica : F
Padrão : 1

KILBUF (0156H)

Função : Limpa o buffer do teclado
Input : Nada
Output : Nada
Modifica : HL
Padrão : 1

CALBAS (0159H)

Função : chama as rotinas do interpretador em qualquer slot
Input : IX - endereço a ser chamado
Output : Depende da rotina chamada
Modifica : Depende da rotina chamada
Padrão : 1

• Novas rotinas do MSX 2

SUBROM (015CH)

Função : Executa uma chamda inter-slot para a subrom
Input : IX - endereço a ser chamado e também o valor de IX no topo da pilha
Output : Depende da rotina chamada
Modifica : AF', BC', DE', HL', IY

EXTROM (015FH)

Função : Executa uma chamada inter-slot para a sub-rom
Input : IX - endereço a ser chamado
Output : Depende da rotina chamada
Modifica : AF', BC', DE', HL', IY

EOL (0168H)

Função : Deleta os caracteres do cursor até o fim da linha
Input : H - coordenada X/ L - coordenada Y
Output : Nada
Modifica : Todos

BIGFIL (016BH)

Função : O mesmo que FILVRM para as novas telas do MSX 2
Input : A=Byte a ser usado pelo FILL, BC=Comprimento em bytes da área a ser preenchida, HL=Endereço onde deve-se começar a preencher
Output : Nada
Modifica : AF,BC,EI

NSETRD (016EH)

Função : Seta a VRAM para leitura de 64K
Input : HL - Endereço da VRAM
Output : Nada
Modifica : AF

NSTWRT (0171H)

Função : Seta a VRAM para escrita de 64K
Input : HL - Endereço da VRAM
Output : Nada
Modifica : AF

NRDVRM (0174H)

Função : Lê um byte na VRAM de 64K
Input : HL - Endereço da VRAM
Output : A - byte lido
Modifica : AF

NWRVRM (0177H)

Função : Escreve um byte na VRAM de 64K
Input : HL - Endereço da VRAM, A - byte a ser escrito
Output : Nada
Modifica : AF

No mês que vem publicaremos as rotinas da sub-rom que sem sombra de dúvida, serão muito importantes para quem deseja desenvolver programas para MSX 2 e MSX 2+. Até lá.

ARTIGO

Construa sua Megaram

R. Pontes
Roberto Silva

A partir desta edição iremos cumprir uma promessa feita pela revista CPU, em uma de suas edições anteriores e que pelo número de cartas enviadas à redação, parece que os usuários estão realmente ansiosos por ela. É claro que só podemos estar falando do projeto da Megaram. O projeto que iremos abordar é de uma Megaram-game de 256Kb que acreditamos satisfazer a vocês, usuários.

Introdução

Assim que o MSX foi lançado no Japão as inúmeras soft-houses japonesas logo viram seu enorme potencial e dentro de pouco tempo já haviam criado uma série de programas para ele e, é claro, jogos, muitos jogos. A tal ponto de o MSX ser confundido por aqui com vídeo-game de luxo.

Para quem ainda não sabe, o Japão é comprovadamente o paraíso mundial dos games. É como se este país estivesse sob alguma espécie de encanto, pois os japoneses são simplesmente fanáticos por jogos e, por isso mesmo, qualquer micro de lá, seja ele MSX, X68000, FM-Towns ou outro modelo qualquer, possua muitos jogos.

Bem parece que esta magia contagiou os micros ocidentais e até mesmo o todo poderoso e burocrático PC já possui um razoável número de games. É até estranho ver um computador que sempre andou de terno e gravata com um ar sério, ser visto por aí desfilando de chinelo, blusão florido e bermudão com aquelas horríveis pernas cabeludas e pálidas por falta de sol. Este fato serve para ilustrar o quanto é rentoso o mundo dos games além de abrir um novo e grande leque de usuários para os micros pessoais. Você deve estar se perguntando o que tudo isto tem haver com a Megaram. A resposta é simples: a Megaram originalmente foi criada para jogos. Os primeiros jogos lançados para MSX no Japão eram vendidos na forma de cartuchos e a sua capacidade era de 16Kb de Rom (memória onde estava contida o jogo). Estes jogos caracterizavam-se pela baixa qualidade gráfica, trilha sonora monótona e fases repetitivas, tais como Road Fighter, Pippols entre outros.

Em seguida, vieram os jogos de 32Kb que, para nós brasileiros, são aqueles jogos de dois blocos do tipo:

```
bload"blabla1.bin",r
bload"blabla2.bin",r
```

Estão lembrados? Nesta categoria encontram-se o Knight Mare, Kung-fu Master etc.

Nestes jogos já se pode notar uma sensível melhoria na parte gráfica, a trilha sonora podia contar com duas ou três musiquinhas diferentes, mas as fases ainda continuavam repetitivas.

Os jogos de 32Kb perduraram por muito tempo, mas como tudo evolui, o mundo dos games não é diferente. Os japoneses sentiram que era necessário a criação de jogos mais sofisticados, com maior qualidade gráfica e trilhas sonoras mais elaboradas, músicas diferentes para cada estágio, bem como estágios completamente diferentes dos anteriores. Mas todos estes requisitos não caberiam em 32Kb ou 64Kb que é o máximo de memória que se pode acessar diretamente em um slot. Para contornar este inconveniente foi desenvolvido o Megarom (acredito ter sido obra da Konami) que pode manipular até 2 Megabytes de memória em apenas um slot.

A Megaram nasceu da dificuldade encontrada em importar do Japão os cartuchos Megarom. Além disso, os poucos cartuchos que aqui chegavam eram vendidos a preços exorbitantes.

A idéia da Megaram é algo muito simples: constituiu-se basicamente na substituição do banco de memória ROM (onde estavam gravados só jogos) por um banco de memória RAM. Com a Megaram criada, foi só retirar os blocos que constituiam cada jogo Megarom e adaptá-los para a Megaram, caso houvesse algum problema de incompatibilidade e criar um programa que se encarregava de procurar a Megaram em qualquer slot do micro, carregar os blocos dos jogos do disquete para a Megaram e finalmente rodá-los. Este programa chama-se MEGARAM.ASM e está presente no carregamento de qualquer jogo de Megaram.

Descrição

A Megaram é um dispositivo que permite ao MSX um acesso a até 2 Megabytes de memória em apenas um slot. Deste total, entretanto, só podemos acessar até 32Kb por vez, isto porque a memória da Megaram está disponível apenas nas páginas 1 e 2 do slot onde a Megaram estiver conectada. Caso o leitor não saiba, o MSX possui quatro páginas de memória (Pg0, Pg1, Pg2, Pg3) sendo que cada uma possui 16Kb.

A Megaram divide as páginas 1 e 2 (no slot onde esta estiver conectada) em duas sub-páginas, criando então 4 sub-páginas de 8Kb cada, a saber:

Sub-página 1.0 - endereços de 4000H a 5FFFH
Sub-página 1.5 - endereços de 6000H a 7FFFH
Sub-página 2.0 - endereços de 8000H a 9FFFH
Sub-página 2.5 - endereços de A000H a BFFFH

O mapeamento da Megaram divide a memória em blocos de 8Kb cada. Como o nosso projeto possui 256Kb, logo temos 256/8 = 32 blocos de memória.

Para cada sub-página da Megaram existe um registro mapeador de 5 bits cada que pode conter um número entre 0 a 31 (estes números correspondem aos 32 blocos de memória da Megaram). Abaixo temos as subpáginas com seus respectivos registros:

Sub-página 1.0 registro mapeador R0
Sub-página 1.0 registro mapeador R1
Sub-página 2.0 registro mapeador R2
Sub-página 2.5 registro mapeador R3

Como o usuário pode perceber, para acessarmos um determinado bloco da memória da Megaram, basta escolhermos em que

subpágina queremos que esta seja acessada e escrever o seu número no registro referente à página escolhida.

Os registros mapeadores (R0 a R3) são registros apenas de escrita, não nos possibilitando ler o conteúdo dos mesmos. Estes registros são acessados pela CPU em certas áreas de memória do slot onde a Megaram estiver conectada e por coincidência é a mesma área de memória gerenciada por cada um deles (observe a tabela 1).

A Megaram dispõe ainda de um quinto registro que chamaremos de registro de acesso. Este registro não existe no original Megarom sendo exclusivo da Megaram.

Em um cartucho Megarom, o jogo está gravado em uma memória ROM que, como já sabemos, é uma memória que só nos permite ler os dados nela contida. Levando-se em consideração isto e que os registros mapeadores (R0 a R3) também são acessados em endereços de memória, apenas de escrita, isto nos leva a conclusão que só existem dois tipos de acesso no Megarom: leitura na memória e escrita nos registros mapeadores.

Na Megaram se fez necessário um terceiro tipo de acesso que é o de escrita na memória, sem o qual não poderíamos carregar a memória da Megaram com um jogo ou outro programa qualquer.

Para possibilitar a escrita em memória foi criado o registro de acesso. Com ele, a Megaram passa a contar com 2 modos de acesso:

Acesso modo 0

Leitura em memória / Escrita nos Regs. mapeadores

Acesso modo 1

Leitura em memória / Escrita em memória

A CPU se liga ao registro de acesso através da porta 8EH. Para mudar o modo do registro de acesso basta uma simples operação de IN ou OUT na porta 8EH ou seja:

OUT (8EH),A - coloca o registro de acesso no modo 0

IN A,(8EH) - coloca o registro de acesso no modo 1

Obs: O valor de A não importa, e sim a operação IN ou OUT em si.

Exemplos de acessos a Megaram

Vamos supor que o usuário queira acessar o bloco de memória da Megaram de número 15 na sub-página 1.5 (6000H a 7FFFH) e que a Megaram esteja conectada no slot 1 (cartridge A no Expert). Para isto devemos escrever o número do bloco de memória no registro mapeador referente a esta subpáginas que é o registro R1.

TABELA 1 – Registros da MEGARAM

Registro mapeador	Endereco de acesso
R0	4000H a 5FFFH
R1	6000H a 7FFFH
R2	8000H a 9FFFH
R3	A000H a BFFFH

1) Já que será feita uma operação de escrita em registro devemos setar o registro de acesso (porta 8EH) no modo 0 (Lê memória / Escreve reg's). Para isto, basta: Out (8EH),A. Lembre-se que o valor contido em A não importa.

2) Colocar a página 1 no slot 1 (slot onde está a Megaram). Isto porque a área de acesso do registro R1 (6000H a 7FFFH) esta "dentro" da página 1 (4000H a 7FFFH).

3) Agora é só escrever o valor 15 em qualquer endereço entre 6000H a 7FFFH que estaremos escrevendo no registro R1.

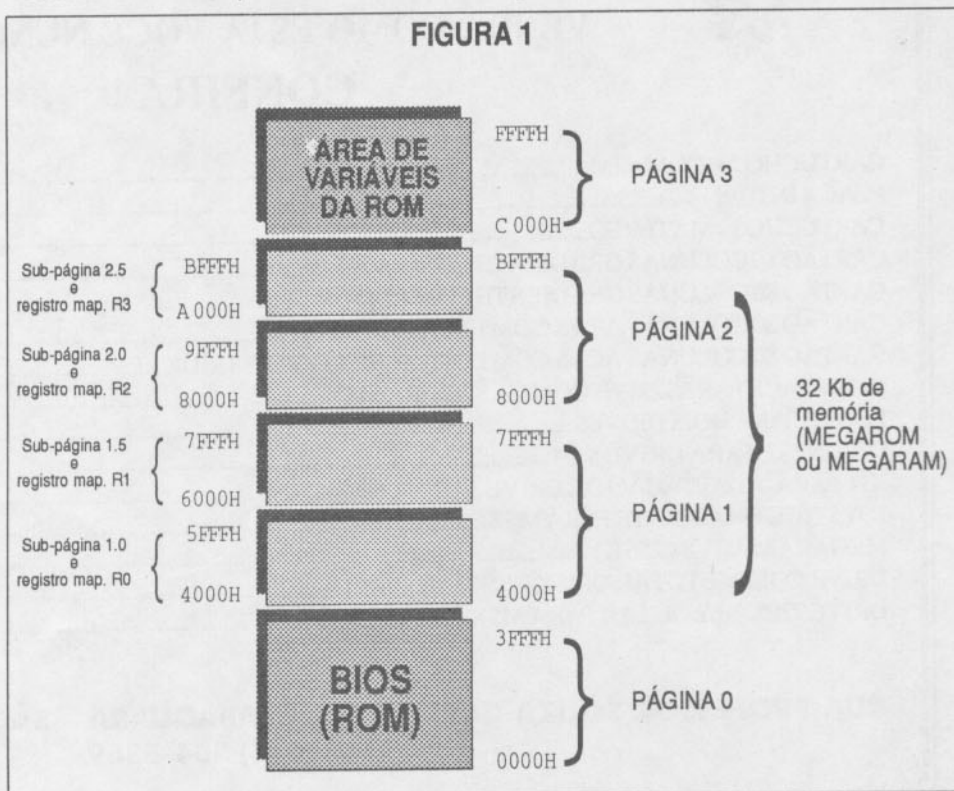
Vamos supor agora que o usuário queira escrever no bloco de memória selecionado acima.

1) Setar o registro de acesso no modo 1 (Lê memória / Escreve memória.): In A,(8EH)

2) Colocar a página 1 no slot 3

3) E então é só escrever entre os endereços 6000H a 7FFFH que estaremos escrevendo no bloco número 15 da memória da Megaram.

Obs: Em qualquer um dos casos acima nos é possível ler na memória da Megaram.

A figura 1 ilustra a paginação de memória com o uso da Megaram. Na figura 2 temos a descrição do circuito na

forma de diagrama em blocos da Megaram game de 256Kb.

O circuito de temporização e controle é o coração da Megaram, pois este gera todos sinais necessários para o acesso a memória e aos registros mapeadores, além de controlar o multiplexador (MPX).

- (Dec) e (F/F access) trabalham em conjunto formando o registro de acesso. Sempre que o Z80 acessa a porta 8EH (Dec) habilita o Filp-Flop (F/F access) a mudar de estado conforme a operação executada, IN ou OUT. Esta mudança é enviada ao circuito de temporização e controle que passa a operar no acesso modo 0 ou modo 1.

- (Reg's Map) são os 4 registros mapeadores. Cada um contém o número de um bloco de memória a ser mapeado. Estes números mais as linhas de endereços da CPU A0 a A12 são enviados ao multiplexador (MPX) que os transformam em linhas de endereços multiplexados MA0 a MA8 para atender ao tipo de endereçamento imposto pelas memórias dinâmicas usadas no banco de RAM.

- O banco de RAM de 256Kb é formado por 8 memórias dinâmicas de 256K x 1 bit (41256).

FIGURA 2 - Diagrama em blocos da Megaram (256 Kb)

Progamação Assembly

O Desafio da Matemática

Julio Cesar Silva Marchi
André Luiz Rocha Tupinambá

Um dos maiores problemas que muitos programadores encontram quanto ao desenvolvimento de programas em assembly é a manipulação de números e equações. A implementação de rotinas que façam cálculos matemáticos, dependendo do nível destes, muitas vezes fazem com que se desista de um projeto depois de inúmeros testes insatisfatórios. Por este e outros motivos, tentaremos demonstrar algumas técnicas usadas em equações de vários tipos, de forma eficiente e compacta (algumas até simples, outras não).

Primeiramente, é fundamental acostumar-se com o fato de que todas as operações dos microprocessadores são em bits. É muito comum os principiantes tentarem fazer com que suas equações sigam o conceito tradicional – isto é, a base decimal – para o reconhecimento dos números, além do processo de somas múltiplas para obter o resultado desejado.

Bem, funcionalmente, isso não está errado. Entretanto, se pararmos para observar, existem técnicas de interações com bits que podem fazer pequenas contas com muito mais velocidade do que se pode imaginar.

Uma das técnicas mais utilizadas é a rotação de bits, que geralmente é utilizada na multiplicação e divisão de um número, tendo como base a potência de dois (2^0=1, 2^1=2, 2^2=4, 2^3=8 etc.). Ao contrário do que possa parecer, este é um processo muito simples de ser implementado em assembly, já que existem inúmeros comandos de rotação de bits.

Para um melhor entendimento, vamos analisar mais profundamente as multiplicações e divisões em assembly:

EXEMPLO 1

Listagem 1

```
MULT10: XOR  A      ; O valo deve entrar em C
        LD   B,10   ; Número de somas
LOOP:   ADD  A,C    ; Soma A com C
        DJNZ LOOP
        LD   C,A    ; O valor retorna também em C
        RET
```

Listagem 2

```
MULT10: LD   A,C    ; 1
        ADD  A,A    ; 2 (ADD A,A tem o mesmo
                    ; efeito de SLA A neste caso,
                    ; só que é mais rápido)
        LD   C,A    ; 3
        ADD  A,A    ; 4
        ADD  A,A    ; 5
        ADD  A,C    ; 6
        RET
```

MULTIPLICAÇÃO

É muito simples entender este conceito: cada bit que se rotaciona para a esquerda realiza uma multiplicação por dois. Assim sendo, consegue-se fazer multiplicações por todas as potências de dois simplesmente rotacionando várias vezes consecutivas o byte para a esquerda. Por exemplo: para multiplicar um número qualquer por oito, devemos fazer três multiplicações por dois (8=2x2x2), o que ficaria da seguinte forma:

```
SLA A    ; SLA rotaciona todos os bits do
         ; registrador para a esquerda
SLA A
SLA A
```

Onde, no registrador A está o multiplicando.

Note que o número de multiplicações por dois é exatamente igual ao logaritmo do número na base dois (log2(8)=3 <=> 2^3=8).

O principal aspecto deste tipo de multiplicação vem do fato de que dela derivam-se todos os outros tipos de multiplicação. Por exemplo, muitas

EXEMPLO 2

Listagem 1:
(Divisão para um número de 16 bits)

```
          ; O número entra em HL e sai em HL

SLA   L   ; Rotaciona os bits de L para a esquerda e
          ; joga o bit 7 em CARRY
RL    H   ; Rotaciona os bits de H e coloca o conteúdo
          ; de CARRY no bit 0 de H
```

Listagem 2:
(Divião para um número de 24 bits)

```
          ; O número entra em DHL e sai em DHL (DHL=registradores D, H
          ; e L combinados)

SLA   L   ; Rotaciona os bits de L para a esquerda e
          ; joga o bit 7 em CARRY
RL    H   ; Rotaciona os bits de H e coloca o conteúdo
          de CARRY no bit 0 de H
RL    D   ; Rotaciona os bits de D e coloca o conteúdo
          ; de CARRY no bit 0 de D
```

pessoas usam uma rotina semelhante a da listagem 1 do exemplo 1 (abaixo) para multiplicar um número qualquer por dez:

Como se pôde observar nitidamente, o valor a ser multiplicado é enviado no registrador C. Se o número enviado fosse por exemplo 3, ele seria incrementado da seguinte forma:

Registrador A	Registrador C
Início0	3
1a3	3
2a6	3
.	
.	
.	
10a30	3

Simples, mas pouco eficiente, esta rotina faz a conta com precisão. No entanto, consome muito tempo (para o assembler é claro).

Observe agora a rotina da listagem 2 do exemplo 1. A princípio esta rotina pode parecer não fazer nada de concreto, mas faz a mesma conta que a anterior só que em muito menos tempo. Para se ter uma idéia, ela realiza a mesma operação 5,47 vezes mais rapidamente, isto é, ela gasta 18,28% do tempo da outra para fazer a mesma conta com a mesma precisão. Muitos podem até dizer que para o assembler esta diferença é insignificante, mas quando esta rotina é chamada várias vezes dentro de um loop, esta diferença se torna substancial, ou então se precisarmos de um cálculo similar entre um acesso e outro a um periférico complexo (como o VDP, drive ou RS232), onde, às vezes, não podemos perder mais do que alguns ciclos de interrupção (ou as vezes não podemos perder nenhum!).

A rotina acima usa o seguinte princípio para o cálculo: você pode transformar uma multiplicação de um número qualquer por uma soma da multiplicação do número por potências de dois. Em linguagem simples, o que ocorre é a aplicação da seguinte propriedade da multiplicação:

$$(A+B)x = Ax + Bx$$

O que fizemos acima foi exatamente isto, separamos o número 10 como uma soma entre 8 e 2. A rotina funciona da seguinte maneira, supondo que se C fosse igual a três:

1º) copia o valor de C em A (A=3, C=3)
2º) multiplica A por dois (A=6)
3º) copia em C o valor de A (C=6)
4º) multiplica A por dois (A=12)
5º) multiplica A por dois (A=24)
6º) soma o valor de A com C
(A=24, C=6 -> A=30)

Tente compreender este princípio, pois dele derivam-se todos os tipos de multiplicação, exatamente a partir deste algoritmo.

DIVISÃO

A divisão por potências de dois é tão simples quanto a multiplicação pelo mesmo método. Enquanto na multiplicação rotacionamos o byte para a esquerda para obter o resultado, na divisão rotacionamos o byte para a direta com aquelas mesmas características da multiplicação.

Para dividir um número por oito, basta rotacionarmos o número para a direita da seguinte forma:

```
SRAA
SRAA
SRAA
```

Até aí tudo bem, mas como a divisão é muito mais complexa do que a multiplicação (estruturalmente falando), existem inúmeros "macetes" para deixar uma rotina de divisão que funciona sobre o método de rotação de bits mais eclética e poderosa e que, principalmente, não oneram a funcionalidade e velocidade da mesma. Um destes "macetes" é usado para obter o resto da divisão. Para tal, proceda da seguinte forma: antes de dividir o número, dê um AND com o divisor menos um. O resultado será o

resto! (Simples, não?) Vamos a um exemplo: para dividir o número 135 por 8 e obter o resto, devemos fazer então:

```
PUSH AF     ; Protege o número
            ; a ser dividido
AND  7      ; Obtém o resto (8-1)
LD   C,A    ; Coloca o resto
            ; em C
POP  AF     ; Recupera o
            ; dividendo
SRA  A
SRA  A
SRA  A      ; Divide por oito
```

DIVISÃO/MULTIPLICAÇÃO DE NÚMEROS DE MAIS DE 16 BITS PELA POTÊNCIA DE DOIS

Um detalhe importante a ser frisado é que nas rotações de bits, temos um limite para a divisão/multiplicação de um número. Se este for de oito bits, o limite máximo são oito interações, se for de dezesseis o máximo serão dezesseis e assim sucessivamente.

Apesar disto, para dividir/multiplicar um número de mais de oito bits não é necessário nenhuma mágica especial, basta apenas dividi-lo em números de oito bits e rotacioná-los a partir do primeiro à esquerda, observando sempre o resultado obtido em CARRY (que é o "macete" desta técnica). Acompanhe as listagens do exemplo 2.

Observe que na listagem 2 do exemplo 2 (divisão por 16 bits), o valor estaria armazenado em três registradores da seguinte forma: DHL.

Outra consideração importante a ser observada é a necessidade de se utilizar as instruções corretas para cada caso. Por exemplo, na multiplicação e divisão de um número de oito bits, pode-se substituir (e deve-se) a instrução SLA r por ADD r,r (vide exemplos anteriores), já que neste caso (e somente neste caso) não precisamos do resultado de CARRY, e, principalmente, porque a instrução SLA r (e similares) gastam de 15 a 23 ciclos de clock para serem executadas, em quanto a instrução ADD r,r gasta somente 4 ciclos. Nas divisões e multiplicações com mais de dezesseis bits, esta substituição torna-se inviável pelo fato de necessitarmos que o conteúdo do bit excedente (bit 0 ou bit 7 dependendo do caso) seja enviado para CARRY, o que é uma particularidade das instruções SLA r e SRA r.

FINALIZANDO

Equações em assembly sempre foram e sempre serão um problema para os programadores. Existem inúmeras técnicas, algoritmos e macetes de programação (alguns gigantescos, mas eficientes) para tornar estas aplicações mais poderosas. Infelizmente, seriam necessárias muitas revistas CPU para entrarmos mais profundamente neste assunto. Entretanto, existem duas rotinas criadas por um desconhecido que foram descobertas por um grande amigo nosso, Julio Velloso, enquanto ele debugava a interface de drive do MSX, uma para divisão e outra para a multiplicação. Ambas operam com números de 16 bits. Estas rotinas foram publicadas em CPU Nº 22 página 44.

Por incrível que pareça, estas rotinas conseguem dividir/multiplicar números de 16 bits, quaisquer que sejam, com uma velocidade impressionante e uma precisão inacreditável, sem contar que a rotina de divisão chega ao cúmulo de retornar o quociente e o resto. O algoritmo utilizado é um pouco complexo e pode ser até caracterizado como maluco, mas o que interessa é que funciona maravilhosamente bem.

Mas, para aqueles mais céticos e que não gostam de ter trabalho (melhor dizendo, MUITO trabalho), aguardem a publicação (em breve) na seção MSX 2 e MSX 2+ desta mesma revista, onde será abordado em detalhes o MATH-PACK do MSX. Para quem não sabe, o MATH-PACK constitui-se em uma série de rotinas existentes na ROM do MSX, que não estão na tabela de JUMPs, mas podem ser chamadas diretamente (fugindo à regra) sem gerar problemas de compatibilidade, já que seus endereços são mantidos como padrão pelos projetistas do MSX. Somente para ilustrar, estas rotinas são as usadas pelo BASIC para gerar suas equações, portanto, todas as contas e equações possíveis de serem feitas em BASIC, também são possíveis em assembly sem muita dificuldade, e, por incrível que pareça, com muita velocidade. Mas não comece a festejar ainda, pois estas rotinas trabalham com uma estrutura numérica conhecida como BCD (Binary Code Decimal). Se você não sabe o que é isso, espere e verá!

Editor de Relógios

Celso Arimatéa Ferreira Junior

Este programa tem por finalidade editar relógios de ponteiros em qualquer parte da tela e em qualquer tamanho. Para isto basta inserir os dados pedidos.

O PROGRAMA

Da linha 10 a 210, o programa pede ao usuário os dados das coordenadas (X e Y) e do raio (R – da linha 40 a 90). Depois ele calcula o maior raio possível de modo que o relógio não vaze a tela). Por fim são pedidos os segundos (S), os minutos (M), as horas (H) e a confirmação destes dados.

Da linha 220 a 360 o programa desenha o relógio. Da linha 370 a 650 o programa o coloca em funcionamento, calculando a rotação dos seus ponteiros, cujas ordenadas são calculadas através da lei dos cossenos – linhas 570 a 650 – e as abscissas, que são calculadas através da lei do triângulo retângulo – linhas 460 a 480 – e o respectivo desenho dos ponteiros gerado na tela – linhas 400 a 440 e de 490 a 510.

COMENTÁRIOS

Este programa não serve somente para criar um relógio de ponteiros. Para o usuário iniciante, a descoberta de novos cálculos poderá ser útil no futuro. Os cálculos trigonométricos que proponho no funcionamento do programa são extremamente simples e de fácil compreensão. Vamos dar um exemplo, supondo que começamos a operar o programa agora:

```
COORDENADA X,Y? 100,90
```

Responda a pergunta como está, ou seja, com as vírgulas. Se isto não for feito, o computador irá obter apenas o valor de "X", fazendo uma outra pergunta para a obtenção do valor de "Y". Após isto, o programa irá apontar um valor máximo para o raio, que será o menor valor entre as coordenadas lançadas e a diferença dos números dispostos na tela e os inseridos (linha 40), diminuído de 16 (na construção do relógio acresce-se no máximo 16 no valor do raio para o compor totalmente – linha 230 a 250). Caso o "T" seja menor que 32 (diâmetro do nosso relógio), o programa irá informar que não existe raio para o relógio.

```
RAIO? 20
```

O valor atribuído ao raio irá orientar o comprimento dos demais ponteiros: 100% deste valor para os segundos (linha 110), 85% para os minutos (linha 140) e 70% para o ponteiro das horas (linha 170).

```
SEGUNDOS? 45
MINUTOS? 30
HORAS? 3.
```

O programa desviará o processamento e fará o cálculo das ordenadas da seguinte maneira: em um relógio convencional, um ponteiro descreve ao longo de sua trajetória um ângulo de 360 graus. Assim, um ponteiro que dá 60 passos (como é o caso do ponteiro dos minutos), cada passo corresponde a seis graus.

Como ângulo formado pelo encontro de dois passos é de seis graus, então a coordenada "Y" deste ponteiro é igual ao seu raio multiplicado pelo cosseno do ângulo (transformado em radianos) multiplicado pelos passos correspondentes até completar 180 graus, com o sinal característico. Isto é a transcrição da regra de três, que reza: $\cos \alpha = Y/R$, com $\alpha=\beta \times 6$, onde β=passos, transformado em radianos, $\alpha = (\beta \times 6) \times$ ¶/180. Até o ângulo

LISTAGEM DO PROGRAMA RELOGIO.BAS

```
10 CLS:KEYOFF:COLOR15,1,1
20 INPUT"COORDENADA X,Y";X,Y
30 IFX<0ORX>255ORY<0ORY>191THEN20
40 T1=255-X:T2=191-Y
50 IFX<YTHENT=XELSET=Y
60 IFT1<TTHENT=T1
70 IFT2<TTHENT=T2
80 IFT<32THENPRINT"O valor destas coordenadas não permitirá a
visão por INTEIRO do relógio.":GOTO100
90 PRINT"O valor máximo do raio será:";T-16
100 INPUT"RAIO";R
110 PI=4*ATN(1):RS=R
120 INPUT"SEGUNDOS (DE 0 A 59)";S:IFS<0ORS>59THEN120
130 GOSUB570
140 RM=R*.85
150 INPUT"MINUTOS (DE 0 A 59)";M:IFM<0ORM>59THEN150
160 GOSUB600
170 TRH=R*.7
180 INPUT"HORAS (DE 1 A 12)";H:IFH<1ORH>12THEN180
190 GOSUB630
200 PRINT"CORRIGIR (S/N)":A$=INPUT$(1)
210 IFA$="S"ORA$="s"THENRUN
220 SCREEN2
230 CIRCLE(X,Y),R+7,14
240 PAINT(X,Y),14
250 CIRCLE(X,Y),R+16,14
```

de 180 graus o sinal é negativo. A partir de então, o passo irá zerar – linhas 580, 610 e 640 –, mas agora o "Y" será positivo. No caso das horas (o ponteiro das horas anda devagar, juntamente com o dos minutos), o cálculo será formado pela posição do ponteiro das horas e dos minutos, tomando-se por base a existência de 60 passos de hora entre uma hora e outra, 720 passos em um ciclo inteiro, perfazendo assim um ângulo de 1/2 graus (aqui representado por ".5" – 360 passos de minuto dividido por 720 passos de horas).

```
CORRIGIR (S/N)? S
```

Neste ponto começa o programa em si, desenhando o relógio e ajustando os ponteiros nos locais predeterminados. Respondendo esta pergunta com um "N" (não), o programa reiniciará.

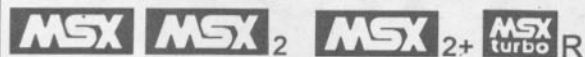
DIGITAÇÃO

A digitação é extremamente simples, cabendo apenas a ressalva dos seguintes detalhes referentes aos símbolos:

- LINHA 330 – LGRA + SHIFT + A
- LINHA 340 – LGRA + A

Estes símbolos poderão ser modificados juntamente com as proporções dos ponteiros. Tente fazer com que, em vez de símbolos gráficos, existam números ou os dois ao mesmo tempo. Dica: use o "loop" da linha 270 e aumente a proporção do relógio nas linhas 80, 90, 250 e 320.

LISTAGEM DO PROGRAMA RELÓGIO.BAS (continuação)

```
260 OPEN"GRP:"AS#1
270 FORF=1TO12
280 IFF>=1ANDF<=6THENG=1:B=F
290 IFF>=7ANDF<=12THENG=-1:B=F-
6
300 W=(R+11)*COS((B*30)*PI/180)
*-G
310 Z=SQR(((R+11)^2)-(W^2))
320 PRESET(X+(G*Z)-3,Y+W-4):COL
OR14
330 IFF=6ORF=12THENPRINT#1,CHR$
(&HFE):GOTO360
340 IFF=3ORF=9THENPRINT#1,CHR$(
&HC4):GOTO360
350 PRINT#1,"."
360 NEXTF:CLOSE#1
370 ONINTERVAL=60GOSUB400
380 INTERVALON
390 GOTO390
400 IFBS=1THENLINE(X,Y)-(PS,QS)
,14
410 IFBM=1THENLINE(X,Y)-(PM,QM)
,14
420 IFBH=1THENLINE(X,Y)-(PH,QH)
,14
430 IFPS=XMANDQS=YMANDBS=1THENL
INE(X,Y)-(PM,QM),1
440 IFPM=XHANDQM=YHANDBM=1THENL
INE(X,Y)-(PH,QH),2
450 BS=0:BM=0:BH=0
460 XS=SQR((RS^2)-(YS^2))
470 XM=SQR((RM^2)-(YM^2))
480 XH=SQR((RH^2)-(YH^2))
490 LINE(X,Y)-(X+(SS*XS),Y+YS),
6:PS=X+(SS*XS):QS=Y+YS
500 LINE(X,Y)-(X+(SM*XM),Y+YM),
1:PM=X+(SM*XM):QM=Y+YM
510 LINE(X,Y)-(X+(SH*XH),Y+YH),
2:PH=X+(SH*XH):QH=Y+YH
520 S=S+1:GOSUB570:BS=1
530 IFS>59THENS=0:M=M+1:GOSUB57
0:BS=1:GOSUB600:BM=1:GOSUB630:B
H=1
540 IFM>59THENM=0:H=H+1:GOSUB60
0:BM=1:GOSUB600:BH=1
550 IFH>12THENH=1:GOSUB630:BH=1
560 RETURN
570 B=S:IFB>=0ANDB<=29THENSS=1
580 IFB>=30ANDB<=59THENB=B-30:S
S=-1
590 YS=RS*COS((B*6)*PI/180)*-SS
:RETURN
600 B=M:IFB>=0ANDB<=29THENSM=1
610 IFB>=30ANDB<=59THENB=B-30:S
M=-1
620 YM=RM*COS((B*6)*PI/180)*-SM
:RETURN
630 B=H:C=M:IFB>=1ANDB<=5THENSH
=1
640 IFB>=6ANDB<12THENB=B-6:SH=-
1
650 IFB=12THENB=0:SH=1
660 YH=RH*COS((B*30+C*.5)*PI/18
0)*-SH:RETURN
```

MSX BITS

Utilizando o Relógio

Miguel Freitas

Muitos colegas programadores e usuários já tiveram problemas para encontrar as horas. Enquanto em algumas versões do DOS e em algumas interfaces de drive o relógio dos micros MSX2 funciona muito bem quando acessado pelo BDOS (endereço 0005H ou F37DH), em outros sistemas nada disso funciona. Algumas documentações técnicas inclusive se referem a estas rotinas como "não implementadas", e dizem apenas que elas não funcionam.

Quando o problema é com a data, tudo bem. Muitas pessoas encontraram o local da memória que mantém a data do computador, de forma que é muito simples ler ou modificá-la. Agora, caso não se consiga acessar a hora do micro pelo modo convencional (DOS/interface), é necessário acessar diretamente a memória estática que existe nos micros MSX2 e isso é feito através de seqüências de OUTs ou por rotinas da SubROM. Não vou me prender ao funcionamento interno da memória estática, que além da data e da hora, armazena também diversas informações como o "password", o país, o prompt e outras.

Os programas listados aqui mostram e modificam as horas, os minutos e os segundos, fazendo o acesso de maneira mais direta possível. Eles podem ser facilmente escritos em outra linguagem que não o Basic e utilizados em qualquer programa.

LISTAGEM 1

```
10 ' Mostra hora no topo da tela
20 OUT &HB4,13:OUT
&HB5,&B1100:CLS
30 LOCATE 0,0,0
40 FOR F = 5 TO 0 STEP -1
50    OUT &HB4,F:IF F=1 OR F=3
THEN PRINT ":";
60    PRINT CHR$((INP(&HB5) AND
&HF) OR &H30);
70 NEXT:GOTO 30
```

LISTAGEM 2

```
10 ' Modifica hora do relógio
20 OUT &HB4,13:OUT
&HB5,&B1100:CLS
30 LINEINPUT"Digite hora
(HH:MM:SS) :";H$:P=1
40 FOR F = 5 TO 0 STEP -1
50    OUT
&HB4,F:A=ASC(MID$(H$,P,1))-&H3060
OUT &HB5,A:P=P+1:IF P=3 OR P=6
THEN P=P+1
70 NEXT
80 GETTIME A$:PRINTA$:'Mostra
as horas p/ conferir
```

BBS: PHOENIX
De: Dimitri Brandi de Abreu
Para: All
Assunto: Leitor OFF-LINE

Alguém que use MSX já ouvir falar de um programa chamado MKQWK, e de algum leitor OFF-LINE para o MSX ?

Estou atrás desse programa para a nossa tão maltratada linha, e qualquer informação será muito bem vinda!

BBS: BDI
De: Carlos Alberto
Para: Dimitri Brandi de Abreu
Assunto: Leitor OFF-LINE

Pelo que vejo você não está habituado a utilizar BBSs, mas vamos ao que interessa:

O MACACA (como o MkQwk é carinhosamente chamado por nós, BBSzeiros), é um programa que roda nos BBSs para possibilitar que você pegue as mensagens que estão nas áreas do BBS em um único arquivo para ler off-line. No MkQwk você deve configurar algumas coisas que estarão envolvidas na transmissão; coisas como o compactador a ser usado, o protocolo de comunicação etc.

Quanto ao leitor off-line, existem alguns que podem ser usados, nesse caso você deve escolher aquele que você melhor se adaptar.

BBS: BDI
De: Henri Michel
Para: All
Assunto: Printing

Salve MSXzeiros em geral! Quem souber como imprimir as telas (os menus) das BBSs, por favor deixe uma mensagem para mim. A propósito: meu modem é o da Gradiente e já vem com o programa de comunicação. Eu sei imprimir no videotexto, mas aqui não funcionou...

BBS: BDI
De: Miguel Freitas
Para: Henri Michel
Assunto: Printing

Bem, você tem algumas alternativas para imprimir: mesmo que o programa da Gradiente não possua este recurso, você pode gravar em um arquivo o texto dos menus e depois imprimir no DOS mesmo. Se o programa da Gradiente também não puder capturar textos, ou você ou instala um programa na memória (se quiser eu tenho ele aqui) para poder imprimir a tela atual pressionando determinada tecla, ou troca logo de programa de comunicação!

BBS: BDI
De: Carlos Alberto
Para: Henri Michel
Assunto: Printing

O programa de comunicação da GRADIENTE (CMM) possui a opção de imprimir. Ela está em uma das teclas de funções, se não me engano é a FX.

Quanto ao problema da impressão, acredito que deve ser algum problema com a sua impressora ou cabos; se não a sugestão do Miguel fica sendo a mais correta: troque de programa de comunicação.

BBS: BDI
De: Carlos Alberto
Para: Miguel Freitas
Assunto: Vírus no MSX

Não é com a intenção de recriminar ou brigar com alguém, mas sim de alertar. Esse negócio de colocar "vírus" em um programa com o propósito de destruir outro, não é uma coisa muito legal não. Já imaginou se todos começarem a colocar vírus (rotinas) nos programas que escrevem para destruir outros que o programador não simpatiza ?

Já pensou se a moda pega?

BBS: BDI
De: Miguel Freitas
Para: Carlos Alberto
Assunto: Vírus no MSX

Calma! Pode deixar que não vou sair fazendo isso! É óbvio que não vou detonar um programa de alguém só porque EU não gosto dele... Assim seria muito fácil, pois os softwares nem iam precisar competir no mercado, um vírus poderia simplesmente apagar o concorrente! :-)

E olha que na história da informática coisas assim já aconteceram, como por exemplo um vírus que atacava apenas os clones dos micros IBMs – se fosse um IBM ele não fazia nada, mas se não fosse... Fique certo que estou ciente dos problemas criados pelos vírus de computador e sei que muitos surgiram entre brincadeiras de programadores, o que, se depender de mim, não vai voltar a acontecer.

Bem, pra falar a verdade, pessoalmente até gostaria de brincar com alguns "viri" criados para o MSX, já que me considero razoavelmente bom entendedor desta máquina e isto não me assusta nem um pouco. Só acho que seria muito chato se um "leigo" fosse contaminado por um vírus, por mais idiota que este seja...

Alguém lembra do famoso "vírus Aquarela"? Vinha um setor no disco original do Aquarela que dizia que ali tinha um vírus que poderia queimar o computador. Confesso que era bem impressionante e nenhum dos meus amigos nunca tentou copiar aquele disquete, mesmo achando o programa muito bom!

BBS: Fantasy
De: Ferrão
Para: All
Assunto: Compactadores

Alguém sabe como descompactar arquivos do tipo .ZIP? Li um artigo na

revista CPU-MSX, mas não vi nada sobre os .ZIP por aqui...

BBS: Fantasy
De: Mario Impronta
Para: Ferrão
Assunto: Compactadores

Para descompactar .ZIP, dispomos do UNZIP099.LZK, que você poderá encontrar aqui mesmo na Fantasy, na área de MSX-compactadores no banco de programas. (infelizmente não temos ainda um ZIPador, mas que sabe um dia desses a gente adapta um...). Mas, voltando ao UNZIP, você precisará descompactá-lo com o UNCR e desarquiva-lo com o LINKER, também disponíveis aqui.

BBS: Macedos
De: Tercio Ferreira
Para: Fernando Carneiro
Assunto: Elite em disco

Tem um amigo meu que tem a versão do jogo Elite que grava em disco. Vou pegar com ele. Acho que não ocupa o disco todo. (A sua versão deve ser "setorizada", não?)

BBS: Macedos
De: Fernando Carneiro
Para: Tercio Ferreira
Assunto: Elite em disco

Ótimo! Vê se arruma a versão com ele, pois a minha é setorizada sim, apesar de termos o diretório, arquivos separados etc.

Verifique, ao gravar os arquivos pelo sistema, se o programa roda corretamente, pois o meu não roda...

BBS: Eduardo
De: Paulo Barreto
Para: All
Assunto: Grupo de usuários

Tenho andado meio afastado e tem diminuído a minha participação na Fido MSX, mas aqui vai meu apelo: o usuário Marcus Vinícius está formando um grupo de usuários e quer entrar em contato com todos os usuários de MSX, especialmente do Grande Rio. Os interessados devem deixar reply a esta msg ou entrar em contato com o Marcus por carta ou telegrama. O endereço é:

Rua Miguel Cervantes, 556 Casa 3
Cachambi – Rio de Janeiro – RJ
CEP: 20780-370

Contamos com a participação de todos.

BBS: FireHawk
De: Erik Kohler
Para: Fernando Carneiro
Assunto: Zmodem

Sobre as rotinas de Zmodem, imagino que você tenha aqueles fontes em Turbo Pascal 6.0 para Zmodem, não é? Seria muito difícil converter para o 3.0? O que uso no lugar de LONGINT? REAL? Ou teria algum meio de simular uma variável LONGINT? E WORD?

BBS: FireHawk
De: Fernando Carneiro
Para: Erik Kohler
Assunto: Zmodem

Bom eu tenho os fontes de umas rotinas de Zmodem aqui sim. Apesar de serem muito grandes, creio que há a possibilidade de adaptá-las para o TP 3.0. Os fontes que possuo parecem estar em TP 4.0, e bem documentados.

Estou tentando entender como funciona o processo do Zmodem para tentar fazer as rotinas em LM, pois ao que me parece estas rotinas em Pascal ficarão grandes após a compilação.

Acredito que a LONGINT seja duas variáveis INTEGER do Pascal 3.0 (4 bytes no total). Nunca cheguei a fazer uma conversão do tipo, mas o meio mais rápido e prático é usar variáveis REAIS.

BBS: FireHawk
De: Fernando Carneiro
Para: All
Assunto: Índios

Num belo dia de sol, o sentinela descobre um bando de índios ferozes se aproximando do forte:

– General, general... Índios à vista !!
– São amigos?
– Devem ser, general. Estão todos juntos!!

ON LINE — MIGUEL DE ANDRADE FREITAS

REDES

Dando continuidade aos esclarecimentos e apresentações a respeito deste incrível mundo das telecomunicações e comunicações de dados, veremos agora as conhecidas redes.

É comum ouvir em um meio de informática, comentários sobre ligações com outras cidades, estados e até países, através de computadores, o que geralmente desperta curiosidade e interesse naqueles que não estão muito por dentro do assunto. As redes de computadores proporcionam isso, que vários computadores, mesmo de países diferentes, estejam ligados, ou se liguem, a fim de trocar informações, programas, mensagens etc.

Essas redes "internacionais" são na minha opinião muito mais interessantes e fascinantes para se comentar do que redes "locais", pois elas abrem aos usuários infinitas possibilidades em todo o mundo, e é de algumas delas, freqüentemente encontradas no Brasil, que vou falar.

FidoNet

Provavelmente esta é a rede mais fácil de se ter acesso que o usuário de MSX pode encontrar. Munido de um modem ele pode falar com cerca de 22.000 BBSs em todo o mundo sem ao menos ligar para fora de sua cidade. É uma rede não-profissional, composta por BBSs voluntárias que não recebem nada por isso, apenas se cadastraram para fazerem parte da FidoNet. Qualquer SysOp pode se cadastrar na rede FidoNet, tendo acesso livre às mensagens de outras BBSs. Como não conta com nenhum tipo de ligação especial, as trocas de mensagens são feitas através de ligações telefônicas normais, geralmente no meio da noite, quando os computadores se ligam sozinhos para botar os "papos" em dia.

Exemplificando o sistema, digamos que você, que acessa uma BBS do interior, envie uma mensagem para uma cidade do interior de outro estado. Para sua mensagem chegar lá acontece basicamente o seguinte: em uma ligação de rotina, a sua BBS passa a mensagem para a BBS coordenadora do seu estado, que por sua vez vai passar para a BBS coordenadora do país, que vai enviá-la ao outro estado para chegar na outra BBS de interior. É claro que este processo não precisa ser exatamente assim, imagine como a BBS coordenadora do Brasil ficaria congestionada, conversando diariamente com todas as coordenadoras dos estados! Ou o "QG" FidoNet, ligando-se com todos os países! É por isso que existem os atalhos, com BBSs de países, estados e cidades diferentes se comunicando diretamente, facilitando em muito o tráfego das mensagens. Apenas para efeito de comparação, segundo o coordenador da região 802 (RJ), Lauro Faria, uma mensagem para chegar ao outro lado do mundo leva em média uma semana.

InterNet

A InterNet já é um pouco mais difícil de se acessar, pois costuma estar presente nas grandes universidades e em centros de pesquisas. Mas muitas pessoas (como eu!) já chegaram nela através de amigos que estudam ou trabalham nestes locais. A InterNet já pode ser considerada uma rede mais "séria", sendo utilizada para a divulgação dos mais diversos temas, além da troca de programas "Shareware" como os anti-vírus. Na InterNet a ligação é feita através de cabos e satélites, que estão permanentemente ligados, possibilitando que um usuário controle um computador do outro lado do mundo como se estivesse na frente dele. Coisas muitos interessantes podem ser encontradas na InterNet, como um enorme banco de programas para CP/M (que podemos rodar perfeitamente no MSX) e documentações diversas. A "megarede" conhecida por InterNet se liga com várias outras redes em diversos países, como a UserNet e a BitNet.

Antes que eu me esqueça: na rede InterNet as comunicações podem ser feitas em até 64kbits por segundo, o que dá pra enviar todo esse texto antes que você consiga falar "todo esse texto"...

* * *

Fire Hawk BBS

Finalmente surgiu "aquela" BBS tão esperada pelos usuários de MSX: Uma BBS feita para e pelo MSX! Enquanto as outras BBSs funcionam em PCs, a Fire Hawk BBS roda em um MSX, com um gerenciador desenvolvido pelo próprio SysOp Fernando Carneiro, também autor do MSX OFFLINE. Por rodar em um MSX, ela possibilita a conexão em uma velocidade muita rara no PC, mas muito comum nos nossos modems: 1200/75: "Apenas" quatro vezes mais rápido do que nas outras BBSs! Confira!

* * *

A seguir vai uma lista de BBSs que possuem áreas de mensagens e arquivos específicos para MSX em todo o país. Continuem a mandar informações sobre outras BBSs com nome, horário, telefone e nome do SysOp.

- **ALFA** (?) tel: (061) 233-0432 SysOp: ?
- **BDI** (24h) tel: (021) 481-1219 SysOp: Lauro Faria
- **Fantasy** (24h) tel: (021) 553-4453 SysOp: Erick Oliveira
- **FireHawk** (21-6h) tel: (021) 393-4490 SysOp: Fernando Carneiro
- **INAMPS** (?) tel: (061) 314-6247 SysOp: Anderson Silva
- **IPI** (24h) tel: (021) 254-3782 SysOp: Anderson Costa
- **Macedo's** (18-8h) tel: (021) 768-5505 SysOp: Marcelo Macedo
- **PHOENIX** (21-7h) tel: (031) 484-1725 SysOp: ?

Videoprodução

MSX e TV... tudo a ver!

Adriano Freitas

Algo curioso que ocorre com relação à linha MSX no Brasil é a falta de informação técnica em determinadas áreas. Talvez isso ocorra devido ao fato de o mercado nacional ser "americanizado", aceitando passivamente os sistemas que nos são impostos pelo marketing americano. Enquanto na Europa o padrão MSX segue a todo vapor – já em sua quarta versão –, aqui está meio estagnado. Muitas pessoas que possuem tal computador não conhecem os recursos que poderia eventualmente dispor. Outras gastam milhares de dólares em um computador para utilizá-lo em tarefas que um MSX (1, 2, 3 ou 4) daria conta tranqüilamente.

Para tentar sanar um pouco da falta de informação quanto aos recursos do MSX 2, inicio aqui esta série de artigos que terá por objetivo dar condições técnicas para que os interessados no assunto criem seus próprios programas. Com isso, quero dizer que tentarei mostrar um pouco das técnicas de desenho/efeitos/animação e a forma como elas podem ser aplicadas ao computador.

As versões do MSX acima da 2.0 possuem recursos bastante razoáveis para utilização em computação gráfica, conseguindo uma relação custo/benefício muito boa, satisfazendo enormemente, por exemplo, um mercado local ou até mesmo regional.

Entretanto, devo ressaltar que, por incrível que pareça, o principal em produção visual não é a mecânica de movimento e sim a criatividade e a harmonia do desenho com as cores que lhe são aplicadas. De nada adianta saber fazer movimentos em perspectivas com desenhos de sólidos e utilizar tal técnica para mover desenhos pobres esteticamente. Pelo contrário, técnicas aparentemente simples, aliadas a um layout bem produzido podem resultar em um visual muito atrativo.

Tendo em vista o que foi exposto acima, nesta série de artigos resolvi omitir as técnicas mais complexas de animação (como movimento de sólidos no espaço), o que resultaria em rotinas complexas em Linguagem de Máquina, o que, aliás, não seria acessível a muitos usuários, uma vez que necessitaríamos, no caso do MSX 2, de um mínimo de 512Kb de memória RAM (Memory Mapper ou Mega RAM). Apresentarei, portanto, efeitos simples, de fácil compreensão e com resultados agradáveis. Não se espante se alguma vez você já utilizou alguma das técnicas com outra finalidade, uma vez que através destes artigos também serão dados "novos usos às velhas rotinas".

A única exigência para que você possa acompanhar esta série é ter domínio sobre a linguagem Basic (não é exigido o mesmo para a Linguagem de Máquina). Quando me refiro ao domínio do Basic incluo os comandos acrescidos ao MSX 2, como COLOR=(n1,n2,n3,n4), SET PAGE, COPY etc.

FERRAMENTAS

Para que possamos trabalhar e/ou estudar computação gráfica, devemos dispor de algumas ferramentas que nos serão úteis para a criação dos layouts, títulos e figuras a serem animadas. Para que você possa acompanhar por completo esta série, sugiro que tenha à mão pelo menos os seguintes programas (o que você possuir além disto será ótimo, afinal cada soft tem seu "ponto forte"):

Criação em Screen 8:
Videographic Philips
Videographic Matsushita

Criação em Screen 7:
Graphic Saurus

Mecânica de efeitos/movimentos:
Um montador Assembler à escolha e o programa TURBO BASIC.

OS NÍVEIS DE CRIAÇÃO

A produção visual (a nível de computação) para TV está basicamente dividida em quatro fases, a saber:

1) **CRIAÇÃO** – O produtor visual define o que será feito em âmbito geral, a duração em segundos e a sonorização a ser utilizada. Definidas estas informações, é desenhado (em papel) um **Storeboard**, que é uma espécie de história em quadrinhos (só que sem os balõezinhos) descrevendo todas as fases da animação.

2) **LAYOUTS** – Seguindo o Storeboard, são criadas separadamente as figuras, títulos, logomarcas, cenários e outros desenhos que posteriormente serão unidos em uma animação.

3) **ANIMAÇÃO** – Onde são dados movimentos e efeitos aos layouts. Os movimentos e efeitos não são produzidos necessariamente de forma contínua, ou seja, em uma vinheta de 30" muitas vezes podem ser feitas de 3 a 10 seqüências, que serão gravadas em separado e posteriormente editadas.

4) **PÓS-PRODUÇÃO** – Esta fase está subdividida em duas outras:

- **EDIÇÃO** – No caso de termos mais de uma seqüência para uma mesma vinheta, é nesta fase que as mesmas são "coladas", dando forma à nossa vinheta.
- **SONORIZAÇÃO** – Nesta fase são colocados os ruídos que caracterizarão e reforçarão os efeitos visuais (por exemplo, quando um raio cai na

tela causando uma explosão seguida do surgimento do logotipo de uma empresa, devemos colocar um ruído característico de um trovão e posteriormente o de uma explosão). É nesta fase que colocaremos também o BGM (Back Ground Music, ou música de fundo). Esta é uma das fases mais importantes da produção visual, afinal, uma seqüência bem sonorizada dá ênfase ao impacto a ser causado pelos efeitos visuais, ao passo que a mesma seqüência incorretamente sonorizada acarretará num empobrecimento dos efeitos, por mais belos que sejam.

PRIMEIROS PASSOS

REGRAS BÁSICAS PARA CRIAÇÃO DOS LAYOUTS

Algumas regras são importantes para aqueles que quiserem produzir algo esteticamente bom para comercialização, apresentação em seminários etc. São elas:

• Jamais se limitar em usar apenas um ou dois softs editores de gráficos. Como já citei, cada editor tem seu ponto forte, portanto, se for necessário, desenhe a tela em etapas, usando o que há de melhor em todos os softs que possuir.

• Como no caso dos editores, não se limite apenas ao uso de uma tela gráfica. Uma boa vinheta pode exigir que você produza parte em baixa resolução e parte em alta. Tudo é muito relativo e você deve saber julgar o que deve ser feito e de que maneira.

• A menos que haja emegência no trabalho, nunca fique longos períodos trabalhando na produção de um cenário ou desenho. Quanto mais você trabalha (seguidamente) com determinado gráfico, mais seus olhos se habituam ao mesmo, tirando-lhe a sensibilidade de saber se o desenho realmente está bom ou não. Um conselho útil é alternar a produção de duas vinhetas ou mesmo duas fases da mesma.

• Seja extremamente crítico com seu trabalho. Não entre naquela de dizer "aquela quininha não ficará melhor do que isso...". Diga: "Tem que ficar melhor!". Somente quando sentir que seu trabalho está perfeito, parecendo feito por profissionais da área, dê por encerrada sua tarefa.

• Recorra sempre que possível aos olhos alheios, afinal, seu trabalho será utilizado para a apreciação dos outros. Aceite sugestões e opiniões de pessoas críticas e sinceras (que não se inibam de dizer que está ruim).

OS TERMOS UTILIZADOS

Existem alguns termos que são utilizados para definir determinadas técnicas, situações e efeitos. As mais utilizadas são as seguintes:

FADE: É conhecida em informática como "tela preta", ou seja, quando temos uma imagem completamente preta sem nenhum um ponto de outra cor, temos um **fade**.

WIPE: É um efeito de troca de imagem e/ou cenário. Por exemplo: quando temos uma imagem, e sobre a mesma começa a se abrir um quadrado dando lugar a outra imagem (que irá tomando todo o espaço do vídeo), temos um **wipe**. Podem existir inúmeros efeitos de Wipes diferentes, como por exemplo, uma imagem substituir outra como se fosse uma cortina se fechando etc.

COR PURA: Todos devem saber que a imagem é formada por uma combinação de pontos que assumem diversas tonalidades de cores. Cada ponto, é um agrupamento de três cores (cores primárias) que juntas resultarão nas outras inúmeras cores. São elas o vermelho, o verde e o azul. Cada uma pode ter sua "quantidade" controlada na mistura, isto é, podemos colocar 3 pontos de azul, 0 de verde e 3 de vermelho para formar a cor violeta. Portanto, são consideradas cores puras as cores primárias, quando as mesmas não apresentam pigmentos de outras cores. Por exemplo a cor formada por Vermelho=5, Verde=0 e Azul=0 é um vermelho puro; Vermelho=0, Verde=7 e Azul=0 também é um verde puro. Já Vermelho=1, Verde=1 e Azul=7 não é considerada uma cor pura, pois apesar de resultar em azul (uma cor primária), possui um pouquinho de verde e vermelho, o que faz com que ela deixe de ser classificada como cor pura.

COR SEMI-PURA: Seguindo a mesma linha de raciocínio das cores puras, devemos apenas acrescentar que a cor **semi-pura** é aquela formada por uma mistura de no máximo 2 cores primárias.

LISTAGEM 1

```
10 '
20 ' MSX E TV... TUDO A VER
30 ' Programa exemplo Nº 01
40 ' Assunto: COLOR MUTARE
50 '
60 '
70 SCREEN 7
80 COLOR 15,1,1
90 CLS
100 IF INKEY$<>CHR$(13) THEN 100
110 FOR F%=1 TO 7:COLOR=(1,0,0,F%):FOR G%=1 TO 10
:NEXT G%:NEXT F%
120 IF INKEY$<>CHR$(13) THEN 120
130 FOR F%=1 TO 7:COLOR=(1,0,F%,7-F%):FOR G%=1 TO
10:NEXT G%:NEXT F%
140 IF INKEY$<>CHR$(13) THEN 140
150 FOR F%=1 TO 7:COLOR=(1,F%,7-F%,0):FOR G%=1 TO
10:NEXT  G%:NEXT F%
160 IF INKEY$<>CHR$(13) THEN 160
170 FOR F%=1 TO 7:COLOR=(1,7,F%,F%):FOR G%=1 TO 1
0:NEXT G%:NEXT F%
180 IF INKEY$<>CHR$(13) THEN 180
190 FOR F%=7 TO 0 STEP -1:COLOR=(1,F%,F%,F%):FOR
G%=1 TO 10:NEXT G%:NEXT F%
200 IF INKEY$<>CHR$(13) THEN 200
```

AGILIZANDO OS PROGRAMAS

Para facilitar a compreensão ao maior número de leitores possível, grande parte dos exemplos serão apresentados em Basic ou, quando for indicado, com uso do Turbo Basic. Logo, precisaremos lançar mão de técnicas que agilizarão o processamento do computador. Uma delas é a utilização, sempre que possível, de variáveis inteiras (aquelas seguidas do sinal de porcentagem). Digite os testes abaixo e compare:

```
TIME=0:FOR F=
1 TO 10000:NE
XT F:PRINT TI
ME\60
```

O computador neste caso, considerará a variável F como sendo uma variável de dupla precisão, contará de um a dez mil e no final apresentará quantos segundos demorou em tal tarefa (cerca de 21 segundos).

```
TIME=0:FOR F%=1 TO 10000:NE
XT F%:PRINT TIME\60
```

Neste exemplo o computador tratará F% como uma variável inteira. E para realizar a mesma tarefa que no exemplo anterior, demorará apenas 9 segundos. Viram como apenas um sinalzinho pode reduzir a velocidade de processamento do computador em até mais da metade?

Utilizaremos muito este "macete", pois trabalharemos bastante com coordenadas e códigos de cores (que são números inteiros). Mas atenção: quando seu programa for utilizar fórmulas para movimentar objetos, construir cenários com fractals etc., você deve evitá-lo – afinal tais fórmulas trabalham com números fracionários e não inteiros. Há, contudo, um recurso que nos permitirá trabalhar com tais fórmulas usando variáveis inteiras, agilizando o processamento, porém descreverei este "macete" mais adiante, quando falarmos de fórmulas matemáticas.

COLOR MUTARE: O PRIMEIRO EFEITO

O primeiro e simplíssimo efeito que aprenderemos será o COLOR MUTARE (mutação de cores). Com ele podemos transformar determinada cor em outra, de forma suave e gradual. Como exemplo, digite o programa da listagem 1.

Ao executar o programa a tela apresentará uma cor, e a cada toque na tecla RETURN a cor atual se "transformará" em uma outra. Isto é COLOR MUTARE. Vamos então, analisar as partes do programa que possam gerar dúvidas.

Como todos devem saber, a Screen 7 trabalha com 16 cores simultâneas de um total de 512 combinações. Através do comando COLOR=(C,R,G,B) definiremos qual tonalidade de cor que cada número entre 1 e 15 representará. Portanto, C

LISTAGEM 2

```
10 '
20 '   MSX E TV... TUDO A VER
30     Programa exemplo Nº 02
40 '   Assunto: Resposta ao exercício 01
50 '
60 '
70 SCREEN 7
80 COLOR 15,1,1
90 CLS
100 R%=7:G=3:B=5:COLOR=(1,R%,G,B)
105 IF INKEY$<>CHR$(13) THEN 105
110 FOR F%=1 TO 5:R%=R%-1:G=G+.8:B=B-.8:COLO
R=(1,R%,G,B):FOR H%=1 TO 10:NEXT H%:NEXT F%
120 IF INKEY$<>CHR$(13) THEN 120
```

será o código da cor que definiremos, R, G e B serão números entre 0 e 7 que dirão qual intensidade de vermelho (Red), verde (Green) e azul (Blue) que deverão ser ativadas nos pontos do vídeo onde teremos a cor de número C.

Quando temos determinada figura no vídeo com a cor 15 codificada em COLOR = (15,7,7,7) e o computador encontra outra ordem informando COLOR = (15,7,0,0), todos os pontos que anteriormente eram branco passarão a ser vermelho, ou seja a troca é instantânea, e não a partir do próximo ponto a ser exibido.

O que o nosso programinha fez foi trocar estes códigos de forma gradual. Veja, por exemplo, a linha 110:

```
110 FOR F%=1 TO 7:COLOR = (
1,0,0,F%):FOR G%=1 TO 10:NE
XT G%:NEXT F%
```

Inicialmente, a cor 1 está codificada da seguinte forma: R=0, G=0, B=0, ou seja, preto. O comando FOR F%=1 to 7 iniciará um rodízio que colocará na variável F% números seqüenciais entre 1 e 7. O comando COLOR=(1,0,0,F%) codificará a cor 1 como sendo R=0, G=0 e B iniciando em 1 e terminando em 7, de acordo com o rodízio produzido pelo FOR. A seqüência FOR G%=1 TO 10:NEXT G% produzirá uma pequena pausa, isto é, fará o computador parar, contar de 1 a 10 e com isso, perder tempo. A partir de agora, passaremos a chamar esta pausa de ESPERA N, onde por exemplo ESPERA 50 seria uma contagem (vazia) entre 1 e 50. Esta perda de tempo é necessária pois senão computador trocaria as cores tão rápido que não nos seria visível. O NEXT F% informará o término do rodízio. O computador então repetirá a seqüência de comandos entre o FOR e o NEXT, contando de 1 a 7 e colocando a cada repetição o valor da contagem em F%. O resultado prático disto será um gradual aumento de intensidade da cor azul, fazendo com que tudo na tela que foi desenhado com a cor 1, que inicialmente era preto se transforme em azul.

Todas as outras linhas que codificam cores seguem o mesmo princípio. Para compreendê-las basta analisar com atenção.

PRIMEIRO EXERCÍCIO

Você deve estar exclamando: "Mais isso qualquer um faz...", então aí vai um pequeno "quebra cuca" para ver se você está mesmo esperto: Utilizando a mesma maneira apresentada no exemplo de COLOR MUTARE, tente fazer um programa que inicialmente apresente no vídeo a cor formada pela combinação R=7, G=3 e B=5 e, ao pressionarmos ENTER, transforme-se em R=2, G=7 e B=1.

Antes de olhar a resposta (listagem 2), tente fazer sozinho. Será um ótimo treinamento, afinal, daqui para frente você deve aprender a achar soluções práticas e eficientes para eventuais problemas.

Antes de analisar a resposta, devo salientar que esta é apenas uma das possibilidades de se resolver o problema. Bem, vamos lá...

Da linha 10 até a linha 90 acredito que ninguém tenha tido dificuldades. Na linha 100, criamos as variáveis R%, G e B com os valores que definirão nossa cor inicial. Não foi erro o fato de colocarmos o sinal de porcentagem apenas na variável R%. Isto ocorreu porque as outras duas, como veremos adiante, trabalharão com valores fracionários. Na mesma linha 100, codificamos a cor 1 como sendo a cor que conterá os parâmetros definidos pelas variáveis R%, G e B. Na linha 105 o computador ficará aguardando o pressionamento da tecla ENTER. Em seguida, iniciamos um rodízio que contará de 1 a 5. Isto porque a variável que sofrerá a maior "transformação" é a R%, que cairá de 7 para 2. Em seguida o número 1 é subtraído da variável R% (isto repetido cinco vezes fará com que o valor da mesma caia de 7 para 2, como solicitado). A variável G é então acrescida de 0.8. Você deve estar se perguntando o por quê de usarmos o valor 0.8. É simples: Como o computador repetirá 5 vezes esta ordem e precisamos aumentar G em apenas quatro pontos, aplicamos a fórmula:

INCREMENTO = AUMENTO / REPETIÇÕES,

Ou seja, o incremento será 4 (o quanto precisamos acrescer à variável) dividido por 5 (número de vezes em que o computador repetirá o comando). Logo, o incremento será de 0.8.

Com a variável B fizemos o mesmo, só que como precisamos reduzí-la e não aumentá-la, diminuimos 0.8 ao invés de somarmos. Acredito que o resto seja de fácil compreensão, dispensando maiores explicações.

USO PRÁTICO

Você deve estar bradando: "Mas computação gráfica não é isso...", o que não é verdade. Apesar de ser um recurso aparentemente banal, na verdade isto é muito útil quando o usamos com criatividade. Veja alguns exemplos:

- Se quisermos fazer o sol nascer, colocamos a "bolinha" do sol surgindo atrás da montanha. Mas só isto não basta para passarmos a ilusão de um nascer do sol. O céu deve acompanhar o movimento, clareando aos poucos. Aí está o uso do COLOR MUTARE.
- Quando temos uma logomarca imitando um material metálico (com brilhos) e queremos dar um movimento a este brilho, reforçando a idéia do metal, também usamos o COLOR MUTARE.

E não termina aí. Cabe somente à sua criatividade saber que efeito usar, em que ocasião e junto com quais outros efeitos. Daqui para frente a complexidade dos exemplos e técnicas irá aumentar gradativamente. No final, você decide como usá-los.

Para finalizar, gostaria de agradecer aos amigos Marco Antônio (o Marcão) e Graça que gentilmente fizeram a revisão técnica do texto.

Um abraço a todos e aguardo vocês na próxima edição de CPU para podermos prosseguir esta série de artigos. Todas as sugestões, críticas e dúvidas serão bem-vindas e, na medida do possível, respondidas.

Nemesis I

Adriano C. R. da Cunha
Vanderlei A. do Nascimento

Nemesis I é um emocionante jogo de ação para seu MSX. Sua missão é enfrentar os Bacterians que invadiram e dominaram oito planetas. Para enfrentá-los, você dispõe da nave Metalion, na qual podem ser acoplados vários tipos de armas.

SEUS INIMIGOS

Existe uma quantidade enorme de inimigos no jogo. Eles estão na superfície, no teto e no espaço dos planetas. Alguns atacarão em uma "formação de ataque". Ao ser destruída esta formação, deixará uma cápsula de energia, que pode ser vermelha ou azul. Alguns tipos de inimigos também deixam cápsulas de energia.

A cápsula vermelha permite que você armazene créditos para a compra de armas. A cápsula azul destrói todos os inimigos da tela.

ARMAS

Suas armas aparecem na parte inferior da tela. Para comprar uma arma, devemos apanhar as cápsulas vermelhas. As armas serão selecionadas seqüencialmente, ficando com a cor amarela a arma atual. Para comprá-la, aperte os segundo botão do joystick ou a tecla "M". São elas:

SPEED UP - Incrementa sua velocidade. Máximo: 6. Muita velocidade é prejudicial, pois seu controle sobre a nave fica difícil.
MISSILE - Dispara misseis na superfície. Ao se chocarem com o chão seguem horizontalmente. Fundamental para eliminar os inimigos terrestres. Máximo: 2.
DOUBLE - Dispara dois tiros, um para frente e outro para cima. Máximo: 1
LASER - É essencial, pois dispara um raio contínuo, destruindo o que estiver na frente. Máximo: 2.
OPTION - É uma sombra que segue sua nave, disparando a mesma arma. Máximo: 2 sombras.
? - É um escudo frontal que é destruído após receber dez tiros.

AS FASES

O jogo é constituído por oito fases. No final de cada fase existe um desafio, e após este ser vencido, você deve destruir a nave-mãe. Para isso, atire em seu centro de força. Cuidado com os seus tiros, senão eles farão churrasco da sua nave.

Sobre as fases, eis as dicas:

FASE 1 - No final da fase existem dois vulcões que explodem quando você chegar. Para não ser atingido, posicione-se no meio deles logo de início.
FASE 2 - Neste mundo existem pedras vermelhas que impedem sua passagem. São destruídas com tiros. Cuidado com os canhões. No desafio, não pare de se mover e atirar.
FASE 3 - É uma das fases difíceis. Atire continuamente na boca dos rostos para destrui-los. No desafio, use a dica da fase 2.
FASE 4 - É igual a fase um, mas os inimigos aumentam. No meio da fase, uma chuva de pedras irá lhe atrapalhar. Fique mais do lado direito da tela, atirando. No desafio, tente acertar as naves quando ainda estiverem em cima da tela.
FASE 5 - É a mais difícil do jogo. Cuidado com os esqueletos e com as caveiras que saem dos poços. Estas podem ser destruídas com tiros. Quase no fim da fase, ossos dançantes tentarão pará-lo. Desvie-se, e, se possuir escudo, use-o para

destruí-los, encostando-o neles. A recomendação do desafio é a mesma da fase 2.

FASE 6 - Também e difícil. Atire no elo vermelho para destruir os tentáculos. O centro resolve-se com alguns tiros. No final existem três monstros com dois tentáculos. Cuidado.

FASE 7 - Não e difícil, pois é semelhante à fase 2. Algumas amebas, quando destruídas, deixam cápsulas de força. Na nave mãe, o uso do escudo e imprescindível.

FASE 8 - A última! Perto do final o caminho estreita-se. Fique no centro, atirando. O uso dos mísseis ajuda bastante. Ao chegar nos braços, entre logo na passagem, se não ela se fechará. O escudo é uma boa defesa. Para destruir o cérebro, atire em suas bases de sustentação.

DICAS

Caso você ache muito difícil o jogo, mesmo com as dicas, digite, após teclar F1, a palavra HYPER, que lhe dará velocidade, escudo, duas sombras, misseis e laser. Cuidado, pois esta dica só pode ser utilizada uma vez no jogo.

Agora, se você é daqueles que não têm paciência de ganhar o jogo "na marra", experimente utilizar o programinha da listagem 1.

Adriano Camargo Rodrigues da Cunha, 8/6/78. Possui um Expert 1.1 com drive 5 1/4, monitor e impressora. Programa em Basic, Assembler e Pascal. Profissão: estudante.

Vanderlei Andrade do Nascimento, possui um Expert 1.1 com drive 5 1/4. Programa em C. Profissão: farmacêutico.

LISTAGEM 1

```
10 DEFINTA-Z:COLOR15,1:SCREEN0:KEYOFF:W
IDTH40:IFFRE(0)<24000THEN70

20 PRINTSPC(16)"NEMESIS":PRINT:INPUT"Vi
das infinitas";V$:INPUT"Armas infinitas
";A$:INPUT"Fase";F

30 BLOAD"NEMESIS.001",R:BLOAD"NEMESIS.0
02",R:BLOAD"NEMESIS.003":POKE&HA56F,F

40 IFV$="S"ORV$="s"THENPOKE&HA3B7,0

50 IFA$="S"ORA$="s"THENPOKE&HA121,0:POK
E&HA192,0

60 DEFUSR=&HD000:X=USR(0):BLOAD"NEMESIS
.004",R

70 PRINT"Tecle CONTROL no BOOT!";INPUT$
(1)

80 DEFUSR=0:X=USR(0)
```

A opção de armas infinitas permite o uso do Hyper quantas vezes você quiser.

Esta dica foi testada em um MSX 1.1 com drive 5 1/4, rodando a versão em disco do jogo Nemesis.

OPERATION WOLF

THE COMPUTER SENSATION OF THE YEAR

Ricardo Jurczyk Pinheiro

INTRODUÇÃO

Para quem viu (e gostou), a série de filmes "Os Doze Condenados" é um dos grandes momentos do cinema. O livro foi bem adaptado e atores como Telly Savallas, Charles Bronson, Lee Marvin, Ernest Borgnine etc., trabalharam muito bem em seus papéis.

A história é "sui generis": durante a Segunda Guerra Mundial, um grupo de doze prisioneiros são treinados para uma missão suicida, como um atentado a um alto general nazista, um ataque a um trem blindado etc. Se eles tiverem sucesso, serão livres. Se fracassarem, serão postumamente homenageados.

Mas vamos ao que interessa. Para quem gosta de filmes (e jogos) de guerra, Operation Wolf é uma grande pedida. Nesse jogo você é um dos doze condenados, com uma missão a cumprir: salvar cinco prisioneiros de campos de concentração nazistas e ajudá-los a fugir. Como a sua vida não é nada fácil, o trabalho sujo sobrou para você: matar o maior número de nazistas possível (incluindo helicópteros, tanques e lanchas) e chegar vivo ao fim da fase. Seria apenas mais um jogo do tipo atire-e-corra, se não houvesse alguns atrativos a mais (como o efeito 3D) e seis fases para "aloprar" a sua cabeça.

AS FASES

As seis fases são: centro de comunicações, selva, paiol, vilarejo, campo de concentração e aeroporto. O jogo só dá apenas um "continue", começando com a munição que você morreu.

O JOGO

Você começa o jogo com sete pentes de bala e cinco granadas. A visão é a dos olhos do jogador e os inimigos correm em quatro planos de visão: perto, meia-distância, longe e no céu. Isso dá uma sensação de tridimensionalidade ao jogador. Não há muito a dizer sobre as fases - apenas atire antes que atirem em você. Seu medidor de ferimentos marca em vermelho (ou violeta) no lado direito da tela seu nível de ferimentos na batalha. Se chegar ao alto da tela, dê adeus a todos e bye-bye! Você foi morto.

MENU PRINCIPAL

Você pode redefinir as teclas. Uma boa idéia é redefinir de modo que a granada seja lançada pela barra de espaços, e depois colocar a opção como Joystick (1). Assim, você controla a mira e atira com o joystick e lança granadas com ACE.

INIMIGOS

Há inimigos para todos os gostos: lançadores de facas, arremessadores de granadas, guardas, soldados,

nazistas rolando, paraquedistas, homens com colete à prova de balas (parecem até com o Schwarzenegger!), tanques, helicópteros etc. Para os nazistas, uma bala bem colocada basta (no caso dos que usam coletes, a bala deve ser na cara!). No caso de tanques e helicópteros, use uma granada, que vai explodir onde estiver o seu cursor.

OBJETOS

De repente, (não mais do que de repente!) caem do ceu certos objetos que podem ser muito úteis. São eles:

• Remédio (Penicilina) - Reduz os ferimentos.
• Pente extra - Mais um pente de balas.
• Aniquilador - Bomba que destrói o que tem na tela.
• Granada extra - Mais uma granada.
• Coxa de galinha - Cai quando você abate um pato selvagem voando no alto da tela. Reduz os ferimentos.
• "SURF" - A sua arma passa a ser "autoshot", sem gastar balas (basta ficar com o botão apertado). Dura enquanto as balas no canto direito da tela ficarem piscando. É ótimo para destruir helicópteros e tanques, pois essas balas podem varam a blindagem dos mesmos. Para ativá-los, basta dar um tiro neles.

DICAS

• A partir da quinta fase, aparecem os prisioneiros, que são cinco na quinta fase e cinco na sexta. Não os mate, mas é quase irresistível atirar em um, pois a morte dele é muito engraçada. Eles atravessam a tela da direita para a esquerda, andando rápido, com as roupas esfarrapadas. Se não forem mortos, agradecem ao sair da tela.
• Para escolher a fase, aperte um número de 1 a 6 quando o jogo estiver carregando. Essa será a fase selecionada.
• Na segunda fase (selva) não há tanques, mas lanchas.
• Se houver helicópteros e tanques na tela, espere o primeiro estar sobrevoando o segundo. Coloque o cursor entre os dois e lance a granada. Adeus tanque e helicóptero!
• Só use as granadas quando for necessário.
• Atire nos paraquedistas, e não nos paraquedas. Se ele estiver "voando baixo", vai cair e sobreviver.
• Atire nas facas e granadas que são jogadas contra você para desviá-las.
• Um bom roteiro é o seguinte: use o mínimo de granadas possível, só matando os soldados e destruindo tanques e helicópteros de vez em quando. Após a morte de todos os soldados, você terá "apenas" a companhia (!) dos tanques e helicóperos. Como você estará cheio de granadas, faça a festa e conclua a fase! Mas não é possível passar de fase, mesmo vencendo-a, já que a cópia que veio para o Brasil está com defeito (assim como a de Double Dragon 2: The Revenge e de Dragon Ninja).

CONCLUSÃO

Se você gosta de jogos de guerra, esse jogo é muito bom (se você não gosta, por que você leu até aqui?). Ele é movimentado, bem desenhado (apesar da carência de cores) e sobretudo rápido e perigoso. Está na minha lista de melhores. Então, um abraço para o major Reissman (Lee Marvin, no segundo filme da série), e outro para vocês leitores. Até a próxima!

DAVID CRANE'S

GHOSTBUSTERS

(OS CAÇA-FANTASMAS)

Maclei dos Santos Tozzato

Caça-fantasmas é um jogo que traz para o seu microcomputador todas as emoções do filme "Os Caças-Fantasmas". Você poderá abrir a sua própria empresa de caça a fantasmas e para isso o banco lhe fará um empréstimo de Cr$ 10.000,00. Ao jogar pela primeira vez, responda "não" ao ser perguntado pelo micro se você tem conta junto ao banco, pois você somente irá obtê-la se conseguir juntar dinheiro para pagar o empréstimo antes que a energia PK da cidade atinja o nível de 10.000. O próximo passo será adquirir equipamentos para a sua empresa. Compre-os de acordo com seu capital. Veja abaixo a descrição e utilidade dos equipamento.

AUTOMÓVEL

Há 4 tipos de veículos que você pode comprar. Observe bem a sua capacidade e o preço antes de fazer a opção de compra.

INSTRUMENTOS DE MONITORAÇÃO

DETECTOR DE ENERGIA PK

Serve para detectar prédios onde há fantasmas e possibilita capturá-los mesmo sem você ter sido chamado. Os fantasmas capturados assim dificilmente valem algum dinheiro, mas é boa publicidade para a sua empresa. Deve ser usado sempre com o intensificador de imagens.

INTENSIFICADOR DE IMAGENS

Ajuda a localizar fantasmas, se usado em conjunto com o detector de energia PK. Caso tenha estes dois equipamentos, fique rodando pela cidade, pois apenas ao passar por um prédio onde existam fantasmas, o equipamento irá mudar de cor. Isto lhe dá a vantagem de não ter que esperar ser chamado para capturar o fantasma, e poderá executar o seu serviço tranqüilamente, fazendo o nome de sua empresa projetar-se na cidade.

SENSOR DE MARSHMALLOW

Se você assistiu ao filme, já sabe que o vilão é um gigantesco monstro de MARSHMALLOW. Quando ele passa pela cidade, dá um enorme prejuízo e quem paga é você, pois ele aparece somente se você não fizer o serviço direito, ou seja, se deixar escapar os fantasmas que encontrar. Este aparelho avisará momentos antes da chegada do monstro. Esteja preparado!

INSTRUMENTOS DE CAPTURA

ARMADILHAS

Obrigatórias para enfrentar os fantasmas – servem para caçá-los. Só podem ser usadas uma única vez para cada fantasma, depois deverá ser levada para a sede de sua empresa para ser rearmada. Você terá que comprar no mínimo uma delas, mas poderá comprar quantas quiser, desde que seu dinheiro seja suficiente e que não provoque excesso de peso no seu automóvel. Se puder comprar o sistema portátil (ver abaixo), irá precisar apenas de uma unidade de armadilha.

ASPIRADOR DE FANTASMAS

Este equipamento é essencial para você capturar os fantasmas das ruas. Funciona somente quando você estiver em trânsito pela cidade.

EQUIPAMENTOS DE ARMAZENAMENTO

SISTEMA PORTÁTIL DE CONFINAMENTO A LASER

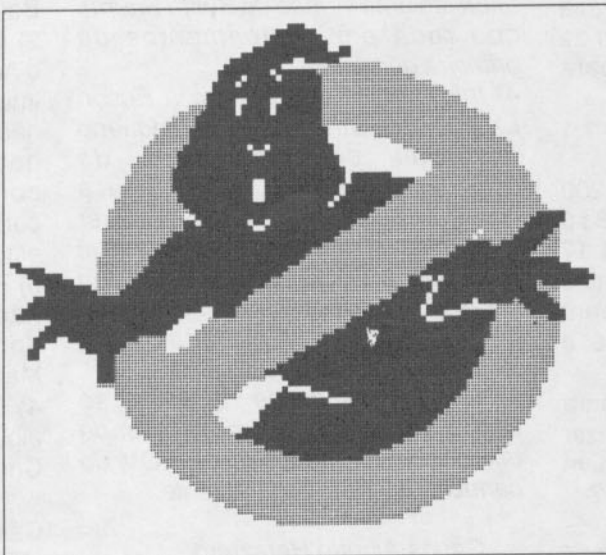
Este é o mais caro equipamento disponível, mas em compensação é o mais útil a sua empresa. Permite que com apenas uma armadilha você possa capturar quantos fantasmas quiser, sem ter que retornar a sede da empresa para prendê-los no locar adequado. Naturalmente você somente poderá comprá-lo depois que tiver sua conta no banco já regularmente aberta.

O JOGO

Após comprar os seus equipamentos, você verá no seu vídeo o mapa da cidade, onde dois prédios estão em destaque. Um deles é a sede da sua empresa (onde estará parado no início) e o outro, bem no centro da cidade, é o edifício do ZUUL. Repare que todos os fantasmas que aparecerem estarão se dirigindo para esse prédio e será lá que você terá que enfrentar o Monstro de Marshmallow (o computador lhe avisará a tempo). Você deverá conquistar a confiança do povo da cidade como eficiente caçador de fantasmas. A pla-

quinha de "Proibido Fantasmas", que é o logotipo de sua empresa, indica a sua posição na cidade e o rastro que ela deixa mostra quando o seu carro terá que andar de um ponto a outro.

Aguarde que um dos prédios comece a piscar (isso representa uma chamada telefônica). Dirija-se então imediatamente para lá e aperte a tecla de tiro. A paisagem agora na tela de seu televisor é de uma estrada, com o seu carro fazendo o trajeto marcado pelos pontinhos do mapa. Durante esse trajeto, você poderá apanhar os fantasmas que perambulam pelas ruas (se tiver comprado o aspirador). Esses fantasmas só aparecem na estrada quando você passa com o logotipo de sua empresa sobre eles no mapa da cidade. Para capturá-los, faça com que o logotipo do carro fique na direção deles e aperte a tecla de tiro.

Quando parar, você estará diante da fachada de um prédio e se houver um fantasma por ali vagando, você terá que capturá-lo. Não se esqueça que cada fantasma que você caçar e não conseguir prender será motivo de descrédito para a sua empresa e poderá provocar a aparecimento do monstro de Marshmallow. Para efetivar a caçada, mova o primeiro homenzinho até que a armadilha que ele carrega esteja em posição de captura (você terá que descobrir a melhor posição de captura). Posicionando o primeiro homem, aperte a tecla de tiro que ele soltará a armadilha no chão. Mova-o então para a posição de combate e tecle de novo o botão de tiro, que um novo homem virá em ajuda do primeiro. Coloque-o também me posição de combate e tecle novamente tiro. Assim, com os dois homens e armadilhas devidamente posicionados, e com o disparo de tiro, seus raios de energia serão lançados contra o fantasma vagante, tentando conduzi-lo o mais próximo da armadilha, quando então, ao seu comando de tiro, ela será ativada e com certeza prenderá o fantasma. Mas tome muito cuidado, pois caso você e sua equipe errem no momento de ativar a armadilha e o fantasma conseguir escapulir, ele dará um susto tão grande em um dos seus homens que será necessário retornar a sede de sua empresa para acalmá-lo. Se você conseguir capturar o fantasma, você ganhará uma recompensa em dinheiro e o seu objetivo é justamente juntar o máximo possível em dinheiro para poder saldar sua conta junto ao banco.

ATENÇÃO
DURANTE O PROCESSO DE CAPTURA,
NÃO CRUZE OS RAIOS!!!

Para conseguir uma conta personalizada junto ao banco, você terá que juntar mais dinheiro do que dispunha inicialmente, e isso antes que a energia PK da cidade atinja o nível de 10.000. Você irá observar dois outros personagens, também importantes para o desenvolvimento do jogo: a chave mestra e o guarda do portal. Quando os dois conseguirem se juntar durante o programa, justamente no edifício do ZUUL, você terá chances de enfrentar o monstro de Marshmallow. Para vencê-lo, basta fazer entrar dois homens no prédio sem que sejam apanhados pelo monstro. Se conseguir tal objetivo, ganhará uma bela recompensa da prefeitura da cidade e muito provavelmente poderá ganhar a sua tão esperada conta personalizada junto ao banco.

Espero que depois de todas essas dicas vocês consigam terminar este sensacional game (na minha opinião). Agora deixemos de papo e mãos a obra!

Inicialmente, gostaria de parabenizá-los pelo excelente nível da revista e pela ajuda já prestada a todos os usuários de MSX.

Gostaria que fossem respondidas as seguintes perguntas:

1) Trocar ou vender programas de linha MSX com outros usuários é crime?
2) O que deve ser feito para comercializar software e hardware da linha MSX, ou simplesmente para abrir um clube para este micro.
3) Existem aplicativos para MSX 1.1 que utilizam o cartucho FM?
4) As impressoras Citizem GX-200 com kit color e Epson LX-810 funcionam perfeitamente no MSX 1.1? Elas imprimem com todos os acentos da língua portuguesa e também gráficos coloridos? Qual delas é a melhor?
5) Existe algum editor de textos para MSX 1.1 com capacidade de utilizar diversos tipos de letras, inclusive negrito, itálico e sublinhado? Qual?

Agradeço desde já pela atenção e sugiro que publiquem mais matérias para MSX 1.1, pois tenho notado há algum tempo a falta de reportagens para essa linha e creio que existam muito mais usuários de MSX 1.1 do que de 2.0, Turbo-R etc.

Jefferson dos Santos – SP

Caro Jefferson,

Do fim para o começo. Sua observação é inteiramente procedente. Mas entenda também o lado dos usuários de MSX 2: quase nada (ou nada mesmo!) havia sido publicado até então! De qualquer jeito, desde a edição passada (35) equilibramos o número de matérias relativas às duas gerações do MSX. Acredite: não é fácil agradar a gregos e troianos!

Quanto às suas perguntas...

1) Se o programa for comercial, ou seja, se alguma empresa ou pessoa detém os direitos de comercialização do mesmo, é crime sim!
2) Para comercializar qualquer tipo de produtos, entre em contato com a distribuidora de determinado software e hardware e faça uma proposta de representação. Já para fundar um clube, basta contar com uma boa biblioteca de softs não-comerciais e nos mandar uma carta. Teremos o maior prazer em divulgar seu clube.
3) Existe uma versão do Synth Saurus que roda em equipamentos de primeira geração.
4) Infelizmente só conheço a Epson LX-810 que, através de um pequeno programa chamado Filtro de acentuação (ver CPU 27), passa a acentuar perfeitamente em qualquer MSX, além de também ser compatível no modo gráfico. Quanto à Citizen GX200, só posso afirmar que será possível acentuar perfeitamente, também com o auxílio do filtro.
5) Com todos esses recursos, só conheço o Astex, um processador de textos que vem gravado na ROM do cartucho CT80-E da Gradiente.

Carlos Alberto Herszteg

• • •

Para começar, gostaria de parabenizar os editores desta revista, que mostra que o MSX ainda não morreu e que ainda existe interesse nesse pequeno, mas poderosíssimo micro.

Sou leitor da revista há pouco tempo, mas gosto muito do trabalho que realizam, desvendando o MSX e preenchendo as informações técnicas não fornecidas pelos livros.

Possuo um MSX 1.1 Plus, com drive de 5 1/4, cartucho Megaram 256K, cartão de 80 colunas, impressora Epson Action Printer T-2000 e vários jogos. Gostaria que vocês sanassem algumas dúvidas que tenho:

1) Pretendo comprar em breve um FM Sound Stereo e mais tarde um MSX 2+. Porém, temo a falta de assistência técnica, uma vez que a Gradiente deixou de fabricar os Experts (perdoem-me se estou dizendo alguma grande besteira). Por acaso comprar uma placa MSX 2+ pode ser comparável a comprar um vídeo Betamax?
2) Quando executo programas como o Aquarela ou King's Valley II, sem ser imediatamente após ligar o micro, eles não funcionam devidamente. Alguém pode me dizer porque? (pode complicar, pois tenho certo conhecimento de Assembler e da estrutura interna do MSX).
3) Por que o Game Master da Konami não funciona para certos jogos Konami de Megaram, como Knight Mare III e King's Valley II?
4) Finalmente, gostaria de obter algumas dicas para La Abadia del Crimen.

Caso alguém se interesse, criei um utilitário que converte telas do Game Master para telas no padrão Aquarela e cria um arquivo SPR para utilizar sprites da Konami como Shapes no Aquarela. Fiz isso porque dispunha de poucos recursos e o Game Master não imprimia telas corretamente na Epson T-2000.

Leonardo A. G. Mesquita – SP

Caro Leonardo

De imediato tenho que concordar fielmente com você em uma coisa: o MSX não morreu e está cada vez com mais força (na Europa e no Japão claro). Aqui no Brasil ele está resistindo com muita garra e tem superado muito bem o marasmo onde se encontrava graças a grupos já não tão isolados que estão lutando pelo MSX, como é o caso da Paulisoft que está trazendo todo mês inúmeras

novidades para MSX da Europa e do Japão (mais até do que o mercado pode consumir).

Estamos tentando suprir a falta de informação técnica, principalmente para dar suporte a programadores (e futuros programadores) para que possam desenvolver projetos sobre o MSX e assim mantê-lo sempre vivo. Acreditamos que os programadores brasileiros, quando com o devido conhecimento, podem ser comparados aos melhores do mundo para o MSX. Para quem não sabe, muitos programas daqui foram pirateados para a Europa e Japão (viu, não somos só nós que pirateamos!), fazendo um grande sucesso, como foi o caso do MSX Page Maker na época do MSX 1, e é o caso hoje do MPW para MSX 2 (nos dias de hoje).

Mas vamos parar de papo furado e voltar ao que interessa. Sobre suas dúvidas:

1) Realmente você não está falando nenhuma besteira, a Gradiente (e a Sharp, que não podemos deixar de citar), abandonaram um projeto que funciona lá fora. O motivo? Não tenho certeza, talvez porque as estimativas de venda não suprissem sua ganância! Mas não adianta ficar especulando sobre isso, o importante é que existem empresas realmente sérias que continuam produzindo e lançando novidades. De duas coisas eu tenho certeza: enquanto existir CPU MSX, informação técnica ao usuário não irá faltar e com certeza empresas interessadas em manter a linha viva, e as empresas que estão aí, vieram para ficar! Portanto, acredito eu que assistência técnica não irá lhe faltar (procure, já que você é de São Paulo, dar um pulinho na Paulisoft e converse com Pepê ou Luciano para dirimir quaisquer de suas dúvidas neste sentido). Observe que o padrão Betamax foi um erro e não vingou em nenhum país, e o MSX não!

2) Não tinha conhecimento deste fato, e apenas com essas informações fica um pouco difícil achar a causa do problema. Provavelmente, este consiste em posicionamento de Slots. Se por acaso estes programas não funcionarem após uma ida ao Basic por exemplo ou após a execução de um programa que efetue o chaveamento de Slots, o problema pode ser este. Por outro lado, é possível que estes programas não tenham sido confeccionados de forma 100% correta, e utilizem-se de recursos inapropriados para máquinas MSX (como acesso direto a rotinas da ROM ou chaveamento da RAM sem procura prévia, etc.). Mas não se assuste, talvez o problema seja simples de resolver! Você falou que manja de assembler, correto? Então pressuponho que conheça as técnicas de chaveamento de Slots e as variáveis de sistema. Estão mãos à obra: crie um programinha que faça com que o seu MSX fique da mesma forma como foi ligado, isto é, Slots, variáveis de sistema e Hooks. Pode não parecer, mas isso é muito simples de se fazer. Ao ligar o micro, observe algumas variáveis de sistemas críticas (como HIMEM (&HFC4A) e SLTATR (&HFCC9) - essa última contém a atual posição dos Slots), e os hooks. Veja se os programas rodam do jeito que está. Se tudo funcionar bem, mãos à obra que provavelmente seu problema estará resolvido.

3) Porque quando estes jogos foram adaptados para MEGARAM, hove alguma alteração a mais que fez com que o Game Master não os reconhecesse como jogos da Konami ou não encontrasse sua tabela de endereçamentos para os dados do jogo ou pior aíida, não encontrasse a STRING de identificação dos jogos (que pode ter sido alterada).

4) Realmente agora você tocou num ponto fortíssimo. Este é o primeiro jogo de MSX (que possui final) ao qual não se há notícias de alguém que o tivesse terminado (talvez o seu criador...). O que sei é que ele segue fielmente a história do filme "O Nome da Rosa", em todos os seus detalhes. Você possui sete dias para solucionar o mistério, após isso você é expulso da sacristia. Você tem que cumprir todos os seus horários de almoço e descanso, assistir às missas, e obedecer aos sacristãos que encontra durante seu trajeto. Primeiro você tem que encontrar um livro (que é quando você vai descobrir a causa das mortes). Para isso, deve-se sair à noite, que é quando as coisas ficam mais fáceis de se solucionar (mas cuidado para não ser pego!). Existem vários objetos espalhados que podem ser pegos e servem para alguma coisa (claro!). Infelizmente só sei até aí, no

mais fica por sua audácia descobrir a solução deste fascinante jogo. Ah, uma dica, na cozinha existe uma passagem secreta...

Quanto ao seu utilitário para conversão de telas do Games Master para o padrão Aquarela, mandenos para uma possível publicação, pois acredito eu que este seja do interesse de muitos usuários.

Bom, acho que já falei muito, portanto um abraço,

Julio Cesar Silva Marchi

• • •

Devido ao bom nível técnico da revista CPU, me espantei ao ver algumas falhas na listagem do programa BATALHA 1917, publicado na CPU número 34, que infelizmente, somente idenfiquei depois de digitar o extenso programa.

A primeira é nada mais nada menos do que a exclusão de uma linha importante do programa, a de número 390.

Como não sou um programador "profissional", me corrijam se eu estiver errado na seguinte suposição: na linha 1944, o programa verifica se duas fórmulas estão se igualando, o que nunca ocorrerá, devido aos seguintes fatos: na segunda fórmula há a soma do valor da variável "RR" com a divisão inteira entre 1 e 32. Mas a divisão inteira entre 1 e 32 é ZERO! Assim, não há número de "RR" que faça com que as duas fórmulas se igualem. Especulei, e acho que a linha 1944 poderá ser:

```
1944 IF (RR+1)/32=((RR+1)\32)
THENGOSUB3920:GOTO2040
```

Assim, os múltiplos de 32 (linha 1943) se unirão aos múltiplos de 31.

Celso Arimatea Ferreira Junior
Rio de Janeiro - RJ

Caro Celso,

É incrível! Por mais que a gente confira, sempre aparece um ou dois errinhos. Você tem toda a razão!

Quanto ao primeiro problema por você apontado, se observar com atenção a posição onde deveria estar a linha 390, vai perceber que ela não desapareceu: apenas "colou" na anterior, sabe-se lá porque. Para corrigir, apague o trecho correspondente a linha 390 e digite novamente esse trecho em uma área limpa da tela. E pronto.

O segundo problema parece que você resolveu. Realmente faz sentido.

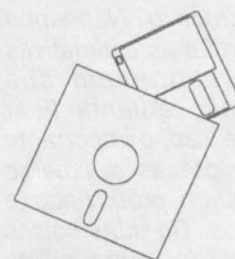
*Acho que é tudo. Desculpe-nos por essas falhas, pois apesar de toda a nossa atenção, **Murphy** é implacável!*

Carlos Alberto Hersz:erg

Em primeiro lugar agradeço por terem publicado o meu editor gráfico na seção "News" da CPU/MSX 34. Gostaria de saber como devo proceder par enviar a listagem de um pequeno jogo, de minha autoria.

Ao ler a CPU/MSX 34, observei a carta de um dos leitores, Alex Assírio Bossi, de Belo Horizonte, e gostaria que vocês publicassem o endereço completo dele ou até que ele mesmo entre em contato comigo, para posterior representação do meu software. Gostaria, se possível, que publicassem o meu endereço para contato junto com meu nome.

Por fim, gostaria de parabenizá-los pelo alto nível da revista.

Alex Assírio Bossi
Rua Cel. Domingos Ortiz, 1075
Franco da Rocha
Centro - São Paulo - SP
Cep 07780-000

TROCAS

Sou possuidor de um MSX EXPERT DD PLUS 1.1 com drive 3 1/2 e gostaria de me corresponder com usuários de

todo país, especialmente com usuários do Rio Grande do Sul que possuam a mesma configuração para troca de programas e informações.

Clovis de Figueiredo
Rua Farroupilha, 361
Centro - Sarandi - RS
Cep 99560-000

• • •

Sou possuidor de um Expert 1.1, drive 5 1/4 de 360 Kb, impressora Lady 80, cartão 80 colunas e monitor monocromático. Tenho dezenas de jogos e aplicativos e gostaria de me corresponder com usuários do MSX para troca de programas, idéias e fazer novas amizades. Gostaria também de me associar a clubes. Todas as cartas serão respondidas.

Benedito Sérgio
Carvalho de Souza
Caixa Postal 2685
Natal - RN
Cep 59022-970

• • •

Possuo um micro Expert 1.1 Br com drive 5 1/4 360Kb e alguns aplicativos e jogos. Sou iniciante na linguagem Dbase II e gostaria de trocar informações e softwares com outros usuários.

Marcus Vinicius dos S. Borges
Rua Cel. José Muniz 714/06
Olinda
26545-080
Nilópolis - RJ

CLUBES

Participe você também do Cabal Soft Club, um clube para MSX que só trabalha com material de domínio publico (nada de pirataria). Veja o que você terá a sua disposição ao se associar:

- Processadores/Editores de texto;
- Ensinamentos de como programar diversos modems
- Programas gráficos;
- Linguagens de programação;
- Possibilidade de obter os programas via modem em velocidades de até 1200/75 CCITT;
- e muito mais ...

Cabal Soft Club
Caixa Postal: 23106
Rio de Janeiro - RJ
Cep 20922-970

DICAS

SUPER MARIO BROSS 2 - MSX 1

Para entrar no mundo 2-2, basta cair no final do mundo 2-1, bem rente a parede de quadrados cinza.

KNIGHT MARE - MSX 1

Para ficar invisível, aperte as teclas SELECT+Y+DIREITA+ESQUERDA várias vezes no início do jogo. Para ficar com 25 vidas, tecle "N" em lugar do "Y".

Ricardo de Magalhães Simões

HINOTORI - MSX 2

Para poder digitar as senhas abaixo, pressione [F1] e em seguida [HOME/CLS] duas vezes. Feito isso, escolha uma senha, digite-a e tecle [ENTER].

METALSLAVE - Ganha 200 penas;
FULLITEMDAYOON - Pega o Mapa e mais seis ítems (o mapa serve para que você possa se deslocar pelas fases);
SUPERBALL - Todas as chaves redondas;
TURBO - Deixa o jogador mais rápido;
GAOOOOOOOOOOH - Ganha 10 vidas (conte as letras O da senha...);
AUTOSHOT - Tiros automáticos;
ILOVENORIKO - Imunidade (mas cuidado para não ficar preso nas encostas!);
ENDDEMOGAMITAINA - Vai para o final do jogo;
ULTRABOX, MAYAME,NANDANANDANANDA, HOIHOIHOINOHOI,KOKMADOKODOKODEMOMAP - Experimente !!!

ZANAC EXCELLENT MSX 2

Para conseguir uma arma secreta, pegue cinco vezes a arma 5 que a sua nave obterá um super-canhão laser.

FEEDBACK - MSX 2

Para acionar o Sound Test, no modo de demonstração pressione as seguintes teclasE+T+C+INSERT até aparecer uma tela totalmente branca, aí solte-as e aguarde o carregamento.

NA PRÓXIMA EDIÇÃO

Rotinas gráficas em C para MSX 2 e MSX 2+

• • •

MEGARAM - Parte final

• • •

Cartas, Jogos, MSX Bits, e muitas novidades...

VALIS II - MSX 2

Para acionar o Soun Teste deste fascinante jogo, pressione [F5] na tela do nome do jogo.

FAMILE BOXING - MSX 2

Experimente a seguinte senha:

AAAAAAPPCPFKELBC

SUPER BOX PRO MSX 2

Para ficar com 99 de energia em todas as opções use o seguinte password: W (sim, só W!).

RAMBO SPECIAL MSX 2

Senhas das fases:

2ª 10A09A38
3ª 04K14042
4ª 03M16041
5ª 05U01K42
6ª 03K12N41
7ª 04K29K38
8ª 03521K40
9ª 12F04M29
10ª 10Y20T31

Obs.: Só funciona nos jogos que não foram mexidos!

Julio Cesar Silva Marchi

GOLVELLIUS - MSX 1

Vá para a opção cotinue e digite o seguinte código o mais rápido possível:

```
6J45L42D23
SFURPK64RT
```

Quando começar o jogo você ficará com:

Find Max: 75000
Find: 37500
Espada - Nível 3 (máximo)
Escudo - Nível 3 (máximo)
Cristal - Todos os sete
Botas - uma para andar na água e outra para dar mais mobilidade
Anel - Para poder destruir as pedras
Colar - Nível 2 (máximo). Aumenta sua resistência aos golpes do inimigo em 4 vezes.

Com esta senha você já começará com a última fase do jogo aberta (esta que se encontra no início do jogo). Para chegar até ela, siga os seguintes passos:

- Ande duas telas à esquerda;
- Suba uma tela;
- Vá mais uma tela à esquerda;
- Suba o rio (isso mesmo, suba pelo rio!);
- Quebre todas as pedras cinza em volta da caverna. Isso é necessário para que você possa quebrar então a pedra que se encontra no meio desta tela, a qual esconde a passagem para a última fase. Boa Sorte!

Nuno Miguel Horta Nunes

AMIGA
FOCUS